Table of Contents

Get Started ................................................................................................................................... 6

Asset Management & Tagging API................................................................. 6

Qualys user account................................................................................................. 7

URL to Qualys API server...................................................................................... 8

Making API calls....................................................................................................... 10

Tracking API usage by user.................................................................................... 14

Know your portal version......................................................................................... 16

Available operators ................................................................................................. 18

JSON Support............................................................................................................. 19

Tags ............................................................................................................................... 23

Create Tag ..................................................................................................................... 23

Update Tag .................................................................................................................... 26

Search Tags................................................................................................................... 29

Count Tags.................................................................................................................... 32

Delete Tag..................................................................................................................... 34

Evaluate Tag............................................................................................................... 36

List Users with their tags.......................................................................................... 39

Tag Fields..................................................................................................................... 43

Get Tag Info............................................................................................................... 44

Host Assets............................................................................................................... 46

Get Host Asset Info .................................................................................................. 46

Create Host Asset .................................................................................................... 55
Update Host Asset ................................................................. 62
Search Host Assets ............................................................. 67
Count Host Assets .............................................................. 86
Delete Host Asset ............................................................... 89
Activate Host Asset ........................................................... 91
Host Asset Fields ............................................................... 95

Assets .................................................................................. 99
Get Asset Info ................................................................. 99
Update Asset ......................................................................... 101
Search Assets ......................................................................... 105
Count Assets ......................................................................... 108
Delete Asset ......................................................................... 110
Activate Asset ......................................................................... 112
Asset Fields ......................................................................... 115

Host Instance Vulnerability data ......................................... 117
Get Vulnerability Info ......................................................... 117
Search Vulnerabilities ......................................................... 119
Count Vulnerabilities .......................................................... 123
Host Instance Vulnerability Fields ....................................... 125

Asset Data Connector ........................................................... 127
Get Connector Info ............................................................. 127
Update Connector ............................................................... 129
Search Connectors .............................................................. 133
Count Connectors.................................................................................................................. 136
Delete Connector ..................................................................................................................... 138
Run Connector ........................................................................................................................ 140
Connector Fields ....................................................................................................................... 143
AWS Asset Data Connector....................................................................................................... 145
Get AWS Connector Info ........................................................................................................ 145
Create AWS Connector ........................................................................................................... 148
Support for AWS GovCloud ................................................................................................. 153
Support for China Region ...................................................................................................... 156
Support for Cross-Account Role Authentication .................................................................. 159
Update AWS Connector ......................................................................................................... 163
Search AWS Connectors ......................................................................................................... 167
Count AWS Connectors .......................................................................................................... 170
Delete AWS Connector ........................................................................................................... 172
Run Connector ........................................................................................................................ 174
AWS Connector Fields ............................................................................................................. 175
AWS Authentication Record .................................................................................................. 177
Get AWS Auth Record Info ................................................................................................. 177
Create AWS Auth Record ..................................................................................................... 179
Update AWS Auth Record ................................................................................................. 181
Search AWS Auth Records ................................................................................................. 183
Count AWS Auth Records ................................................................................................. 186
Delete AWS Auth Record ..................................................................................................... 188
AWS Auth Record Fields ................................................................. 190
Get Started

Asset Management & Tagging API

Manage assets in your account that you want to scan for security and compliance, define asset tags and AWS connectors.

Modules supported

VM, PC, SCA

Authentication

Authentication to your Qualys account with valid Qualys credentials is required for making Qualys API requests to the Qualys API servers. Learn more about authentication to your Qualys account

Get API Notifications

We recommend you join our Community and subscribe to our API Notifications RSS Feeds for announcements and discussions.

https://community.qualys.com/community/developer/notifications-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 Cloud Apps deliver businesses critical security intelligence continuously, enabling them to automate the full spectrum of auditing, compliance and protection for IT systems and web applications on premises, on endpoints and elastic clouds. For more information, please visit www.qualys.com

Qualys and the Qualys logo are proprietary trademarks of Qualys, Inc. All other products or names may be trademarks of their respective companies
Qualys user account

Authentication to your Qualys account with valid Qualys credentials is required for making Qualys API requests to the Qualys API servers.

The application must authenticate using Qualys account credentials (username and password) as part of the HTTP request. The credentials are transmitted using the “Basic Authentication Scheme” over HTTPS.

For information, see the “Basic Authentication Scheme” section of RFC #2617:

http://www.faqs.org/rfcs/rfc2617.html

The exact method of implementing authentication will vary according to which programming language is used.

The allowed methods, POST and/or GET, for each API request are documented with each API call in this user guide.

Sample request - basic authentication

```bash
curl -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/qps/rest/2.0/count/am/hostasset
```
URL to Qualys API server

Qualys maintains multiple Qualys Cloud Platforms. The API server URL that you should use for API requests depends on the platform where your Qualys account is located.

<table>
<thead>
<tr>
<th>Account location</th>
<th>API server URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualys US Platform 1</td>
<td><a href="https://qualysapi.qualys.com">https://qualysapi.qualys.com</a></td>
</tr>
<tr>
<td>Qualys US Platform 2</td>
<td><a href="https://qualysapi.qg2.apps.qualys.com">https://qualysapi.qg2.apps.qualys.com</a></td>
</tr>
<tr>
<td>Qualys US Platform 3</td>
<td><a href="https://qualysapi.qg3.apps.qualys.com">https://qualysapi.qg3.apps.qualys.com</a></td>
</tr>
<tr>
<td>Qualys EU Platform 1</td>
<td><a href="https://qualysapi.qualys.eu">https://qualysapi.qualys.eu</a></td>
</tr>
<tr>
<td>Qualys EU Platform 2</td>
<td><a href="https://qualysapi.qg2.apps.qualys.eu">https://qualysapi.qg2.apps.qualys.eu</a></td>
</tr>
<tr>
<td>Qualys India Platform 1</td>
<td><a href="https://qualysapi.qg1.apps.qualys.in">https://qualysapi.qg1.apps.qualys.in</a></td>
</tr>
<tr>
<td>Qualys Private Cloud Platform</td>
<td><a href="https://qualysapi">https://qualysapi</a>.&lt;customer_base_url&gt;</td>
</tr>
</tbody>
</table>

Looking for your API server URL for your account? You can find this easily. Just log in to your Qualys account and go to Help > About. You’ll see this information under Security Operations Center (SOC).
Making API calls

Curl samples in our API doc

We use curl in our API documentation to show an example how to form REST API calls, and it is not meant to be an actual production example of implementation.

Object types

You have core objects, which represent domain objects for specific business goals and related objects which contain related information or collections of information. Related objects are often simplified representations of core objects but are not implicitly core objects. For example, the tags collection on Asset is a simpler form of the Tag core object, but the ports collection is not.

Collections

Collections of related objects are found within a container object called a QList. These lists will have a specific name for the type of objects they contain. For example, the tags collection Asset is a TagSimpleQList and will read and write TagSimple API objects. These lists can contain a number of sub elements.

- count - (Read only) The total number of items returned in the list element
- list - (Read only) The items contained in the collection on the server
- set - A new collection of items to place in the server side object. Any existing items not in the list provided will be discarded.
- add - A new item to be added to the server side object. The item may be keyed of one ore more fields depending on the collection. In the even that that an item in the add collection collides with an existing entry, the existing entry will be updated with the fields provided. Many collections will allow you to either associate an existing item with the targeted collection, or create a new one and add it to the collection. If you provide a key field, most often id or uuid, the object will be looked up and associated. In the absence of these fields, a new object will be created (if the list allows it).
remove - Removes an element from the list by the collections key, usually id. If the item does not exist, the entry will be ignored. Additional fields beyond the item key will also be ignored.

update - Updates item(s) in the collection. This allows you to update the fields of non-core items via the objects and reference them. Items will be resolved by the collection’s key, and then additional fields applied to the found object. In the event that the supplied item does not match an existing related object, it will be ignored.

Whitespace in HTML tags

Whitespace (which includes line breaks) is not allowed in XML tags that are numbers.

<table>
<thead>
<tr>
<th>Invalid tag - This syntax will not work</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;34234&lt;/id&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Valid tag - This syntax will work just fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;345254&lt;/id&gt;</td>
</tr>
</tbody>
</table>

Pagination

Some API actions will return a list of core objects but will limit the number returned (default is 100). You can change which objects are returned and the number of objects by specifying a preferences tag in the POST body of your request.

Preferences tag fields:

startFromOffset - The first item to return by index. The default is 1.

startFromId - The first item to return by primary key. No default value.

limitResults - The total number of items to return. The default is 100.
The allowed methods, POST and/or GET, for each API request are documented with each API call in this user guide.

### Sample pagination settings

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <preferences>
    <startFromOffset>100</startFromOffset>
    <limitResults>50</limitResults>
  </preferences>
</ServiceRequest>
```

### Limit your results

Use the optional “fields” parameter for any Search or Get API request to limit the amount of information returned in the results. Simply specify the fields you want to include in the output, and all other information will be filtered out. Multiple fields are comma separated and wildcards are supported.

This get request will fetch tag ID 12345 and return the tag ID, name and creation date:

### Sample limit results

```
https://qualysapi.qualys.com/qps/rest/2.0/get/am/tag/12345?fields=id,name,created
```

This search request will return the ID of the connector and the ID of any default tags attached to the connector:

### Sample search connectors

```
https://qualysapi.qualys.com/qps/rest/2.0/search/am/awsassetdataconnector?fields=id,defaultTags.list.SimpleTag.id
```

Using wildcards, the example above could be represented as:

### Sample search connectors using wildcards
https://qualysapi.qualys.com/qps/rest/2.0/search/am/awsassetdataconnector?fields=id,defaultTags.*.*.iddSimpleTag.id
**Tracking API usage by user**

You can track API usage by a user without the need to provide user credentials such as the username and password.

**Optional X-Powered-By header**

API usage can be tracked using the X-Powered-By HTTP header which includes a unique ID generated for each subscription and a unique ID generated for each user. Once enabled, the X-Powered-By HTTP header is returned for each API request made by a user. The X-Powered-By HTTP header will be returned for both valid and invalid requests. However, it will not be returned if an invalid URL is hit or when user authentication fails.

Contact Qualys Support to get the X-Powered-By HTTP header enabled.

The X-Powered-By header is returned in the following format:

```
X-Powered-By: Qualys:<POD_ID>::<SUB_UUID>::<USER_UUID>
```

where,

- POD_ID is the shared POD or a PCP. Shared POD is USPOD1, USPOD2, etc.
- SUB_UUID is the unique ID generated for the subscription
- USER_UUID is the unique ID generated for the user. You can use the USER_UUID to track API usage per user.

**Sample X-Powered-By header**

```
X-Powered-By: Qualys:QAPOD4SJC:f972e2cc-69d6-7ebd-80e67b9a931475d8:06198167-43f3-7591-802a-1c400a0e81b1
```

**Sample outputs**

Here are sample outputs showing the X-Powered-By HTTP header.

**Sample output for VM, PC**
Sample output for other Qualys apps

229 HTTP/1.1 200 OK
X-Powered-By: Qualys:QAPOD4SJ:CA903-a9e1-8b-8893
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Mon, 04 Dec 2017 05:36:29 GMT
Server: Apache
LBDEBUG: NS=10.44.1.12,SERVER=10.44.77.81:50205,CSW=cs-p04-qualysapi443,VSERVER=vs-p04-papi-80,ACTIVE-SERVICES=2,HEALTH=100
Know your portal version

/qps/rest/portal/version/

[GET] [POST]

Using the Version API you can find out the installed version of Portal and its sub-modules that are available in your subscription.

Sample XML

**API request**

```
```

**Response**

```
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Portal-Version>
      <WAS-VERSION>6.0.0.0</WAS-VERSION>
      <FIM-VERSION>1.5.1</FIM-VERSION>
      <VM-VERSION>1.0.3</VM-VERSION>
      <CERTVIEW-VERSION>1.1.0.0</CERTVIEW-VERSION>
      <CM-VERSION>1.20.1</CM-VERSION>
      <MDS-VERSION>2.11.7.0</MDS-VERSION>
      <CA-VERSION>2.9.1.0</CA-VERSION>
      <IOC-VERSION>1.1.0</IOC-VERSION>
      <AV2-VERSION>0.1.0</AV2-VERSION>
      <QUESTIONNAIRE-VERSION>2.14.0.4</QUESTIONNAIRE-VERSION>
      <WAF-VERSION>2.7.0.0</WAF-VERSION>
    </Portal-Version>
  </data>
</ServiceResponse>
```
Sample JSON

API request


Response

```json
{  "ServiceResponse": {  "data": [  {  "Portal-Version": {  "PortalApplication-VERSION": "2.33.0.0-SNAPSHOT-1 DEVELOP #352 (2018-05-07T22:53:43Z)",  "WAS-VERSION": "6.0.0.0",  "VM-VERSION": "1.0.3",  "CM-VERSION": "1.20.1",  "MDS-VERSION": "2.11.7.0",  "CA-VERSION": "2.9.1.0",  "QUESTIONNAIRE-VERSION": "2.14.0.4",  "WAF-VERSION": "2.7.0.0"  },  ...  ]  },  "responseCode": "SUCCESS",  "count": 1  }
```
Available operators

Operators supported by input parameters:

Integer - EQUALS, NOT EQUALS, GREATER, LESSER, IN

Text - CONTAINS, EQUALS, NOT EQUALS

Date - EQUALS, NOT EQUALS, GREATER, LESSER

Keyword - EQUALS, NOT EQUALS, IN

Boolean (true/false) - EQUALS, NOT EQUALS

* NOT EQUALS operator is not supported for update and delete actions. Using the NOT EQUALS operator for updating or deleting objects (such as tags, assets, host assets, AWS connectors, AWS authentication records, etc.) could result in accidental update or deletion of the objects without any warning. To prevent accidental updates/deletions, we do not support NOT EQUALS operator for updating/deleting objects.
JSON Support

Qualys Asset Management and Tagging API supports JSON requests and responses starting with version 2.11. Samples are shown below.

<table>
<thead>
<tr>
<th>Headers used in samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send JSON request</td>
</tr>
<tr>
<td>&quot;Content-Type: application/json&quot;</td>
</tr>
<tr>
<td>Get response in JSON</td>
</tr>
<tr>
<td>&quot;Accept: application/json&quot;</td>
</tr>
</tbody>
</table>

Sample 1 - Create a tag

**API request**

```bash
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/tag"
```

POST data:

```json
{  
  "ServiceRequest": {  
    "data": {  
      "Tag": {  
        "name": "Parent Tag",  
        "ruleType": "NAME_CONTAINS",  
        "ruleText": "windows",  
        "color": "#FFFFFF",  
        "children": {  
          "set": {  
            "TagSimple": [  
              { "name": "Child 1" },  
              { "name": "Child 2" }  
            ]  
          }  
        }  
      }  
    }  
  }  
}
```
XML output

```xml
{
  "ServiceResponse" : {
    "data" : [ {
      "Tag" : {
        "ruleText" : "windows",
        "color" : "#FFFFFF",
        "modified" : "2016-01-04T19:51:56Z",
        "name" : "Parent Tag",
        "children" : {
          "list" : [ {
            "TagSimple" : {
              "name" : "Child 2",
              "id" : 2066216
            }
          }, {
            "TagSimple" : {
              "name" : "Child 1",
              "id" : 2066217
            }
          ]
        },
        "created" : "2016-01-04T19:51:56Z",
        "ruleType" : "NAME_CONTAINS",
        "id" : 2066215
      }
    },
    "count" : 1,
    "responseCode" : "SUCCESS"
  }
}
```

Sample 2 - Search tags

API request

POST data:
{
   "ServiceRequest": {
      "filters": {
         "Criteria": [{
            "field": "parent",
            "operator": "EQUALS",
            "value": "2035617"
         },
         {
            "field": "name",
            "operator": "CONTAINS",
            "value": "child"
         },
         {
            "field": "id",
            "operator": "IN",
            "value": "2035619,2035618,2029815"
         },
         {
            "field": "ruleType",
            "operator": "EQUALS",
            "value": "GROOVY"
         },
         {
            "field": "color",
            "operator": "EQUALS",
            "value":="#EC7000"
         }
      }
   }
}
"ServiceResponse" : { 
  "data" : [ { 
    "Tag" : { 
      "ruleText" : "windows",
      "color" : "#FFFFFF",
      "modified" : "2016-01-04T19:51:56Z",
      "name" : "Parent Tag",
      "children" : [ 
        { 
          "TagSimple" : { 
            "name" : "Child 2",
            "id" : 2066216
          }
        }, 
        { 
          "TagSimple" : { 
            "name" : "Child 1",
            "id" : 2066217
          }
        } 
      ],
      "created" : "2016-01-04T19:51:56Z",
      "ruleType" : "NAME_CONTAINS",
      "id" : 2066215
    }
  } ],
  "count" : 1,
  "responseCode" : "SUCCESS"
}
Tags

Create Tag

/qps/rest/2.0/create/am/tag

[POST]

Create a new tag and possibly child tags.

Permissions required - Managers with full scope, other users must have these permissions: Access Permission “API Access”, Tag Permission “Create User Tag”, Tag Permission “Modify Dynamic Tag Rules” (to create a dynamic tag)

Sample - Create new tag with 3 child tags

API request

curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/tag" < file.xml

Note: “file.xml” contains the request POST data.

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <Tag>
      <name>Parent Tag</name>
      <ruleType>Groovy</ruleType>
      <ruleText>if(asset.getAssetType()!=Asset.AssetType.HOST) return false;
return asset.hasVulnsWithSeverity(4,5)</ruleText>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
      <color>#FFFFFF</color>
      <children>
        <set>
<TagSimple>
  <name>Child 1</name>
</TagSimple>
<TagSimple>
  <name>Child 2</name>
</TagSimple>
<TagSimple>
  <name>Child 3</name>
</TagSimple>
</set>
</children>
</Tag>
</data>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Tag>
      <id>1589217</id>
      <name>Parent Tag</name>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
      <color>#FFFFFF</color>
      <ruleText>if(asset.getAssetType()!=Asset.AssetType.HOST) return false;
      return asset.hasVulnsWithSeverity(4,5)</ruleText>
      <ruleType>GROOVY</ruleType>
      <children>
        <list>
          <TagSimple>
            <id>1</id>
            <name>Child 1</name>
          </TagSimple>
          <TagSimple>
            <id>2</id>
            <name>Child 2</name>
          </TagSimple>
        </list>
      </children>
    </Tag>
  </data>
</ServiceResponse>
Get to know Groovy

Check out the following article on our Community to learn how to create asset tags using the Groovy programming language. You’ll also get several Groovy rule examples that you can start using today.

[Create Asset Tags using Groovy](#)
Update Tag

/qps/rest/2.0/update/am/tag/<id>
/qps/rest/2.0/update/am/tag

[POST]

Update fields for a tag and collections of tags.

Click here for available operators

Using the NOT EQUALS operator for updating tags could result in accidental update of unknown tags without any warning. To prevent accidental updates of unknown tags, we do not support NOT EQUALS operator for update actions.

Permissions required - Managers with full scope, other users must have these permissions: Access Permission “API Access”, Tag Permission “Create User Tag”, Tag Permission “Modify Dynamic Tag Rules” (to create a dynamic tag)

Sample - Rename parent tag, remove some child tags

API request

```
```
Note: “file.xml” contains the request POST data.

Request POST data

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <Tag>
      <name>Parent Tag (Updated)</name>
      <children>
        <remove>
          <TagSimple><id>123</id></TagSimple>
```

26
<TagSimple><id>456</id></TagSimple>
</remove>
</children>
</Tag>
</data>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Tag>
      <id>12345</id>
      <name>Tag</name>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
      <color>#FFFFFF</color>
      <ruleText>asset.installedSoftwares.contains { it.name == "Windows" }</ruleText>
      <ruleType>GROOVY</ruleType>
      <children>
        <list>
          <TagSimple>
            <id>1</id>
            <name>Child 1</name>
          </TagSimple>
          <TagSimple>
            <id>2</id>
            <name>Child 2</name>
          </TagSimple>
          <TagSimple>
            <id>3</id>
            <name>Child 3</name>
          </TagSimple>
          <TagSimple>
            <id>123</id>
            <name>Linked Child 1</name>
          </TagSimple>
        </list>
      </children>
    </Tag>
  </data>
</ServiceResponse>
<id>456</id>
  <name>Linked Child 2</name>
</TagSimple>
</list>
</children>
</Tag>
</data>
</ServiceResponse>
<responseCode>SUCCESS</responseCode>

XSD

<platform API server>/qps/xsd/2.0/am/tag.xsd
Search Tags

/qps/rest/2.0/update/am/tag/<id>
/qps/rest/2.0/update/am/tag

[POST]

Returns a list of tags that match the provided criteria.

Pagination - A maximum of 100 tags are returned by default. To customize this specify a “preferences” tag in the POST body of your request.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the tag. Learn more

Permissions required - Managers with full scope, other users must have Access Permission “API Access”

Searchable Fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (Integer)</td>
<td></td>
</tr>
<tr>
<td>name (Text)</td>
<td></td>
</tr>
<tr>
<td>parentTagId</td>
<td></td>
</tr>
<tr>
<td>parentTagId (Integer)</td>
<td></td>
</tr>
<tr>
<td>ruleType</td>
<td>STATIC, GROOVY, OS_REGEX, NETWORK_RANGE, NAME_CONTAINS, INSTALLED_SOFTWARE, OPEN_PORTS, VULN_EXIST, ASSET_SEARCH, CLOUD_ASSET</td>
</tr>
<tr>
<td>color</td>
<td>Text formatted as #FFFFFF where F can be any value</td>
</tr>
</tbody>
</table>
Sample - Find tags with groovy script rules

**API request**

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/tag" < file.xml
```

Note: “file.xml” contains the request POST data.

**Request POST data**

```xml
<ServiceRequest>
  <filters>
    <Criteria field="ruleType" operator="EQUALS">GROOVY</Criteria>
  </filters>
</ServiceRequest>
```

**Response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <data>
    <Tag>
      <id>12345</id>
      <name>Tag</name>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
      <color>#FF0000</color>
      <ruleText> asset.installedSoftwares.contains { it.name == "Windows" } </ruleText>
      <ruleType>GROOVY</ruleType>
      <children>
        <list>
          <SimpleTag>
            <id>123</id>
          </SimpleTag>
        </list>
      </children>
    </Tag>
    <Tag>
      <id>12345</id>
      <name>Tag</name>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
      <color>#FF0000</color>
      <ruleText> asset.installedSoftwares.contains { it.name == "Windows" } </ruleText>
      <ruleType>GROOVY</ruleType>
      <children>
        <list>
          <SimpleTag>
            <id>123</id>
          </SimpleTag>
        </list>
      </children>
    </Tag>
  </data>
</ServiceResponse>
```
<name>Red</name>
</SimpleTag>
</list>
</children>
</Tag>
<Tag>
<id>12346</id>
<name>Another Red Tag</name>
<created>2014-02-06T19:14:50Z</created>
<modified>2014-02-06T19:14:50Z</modified>
<color>#FF0000</color>
<ruleText>asset.installedSoftwares.contains { it.name == "Windows" }</ruleText>
<ruleType>GROOVY</ruleType>
<children>
<list>
<SimpleTag>
<id>123</id>
<name>Red</name>
</SimpleTag>
</list>
</children>
</Tag>
</data>
</ServiceResponse>
<responseCode>SUCCESS</responseCode>

XSD

<platform API server>/qps/xsd/2.0/am/tag.xsd
Count Tags

/qps/rest/2.0/count/am/tag

[POST]

Count all the children of a tag.

Permissions required - Managers with full scope, other users must have Access Permission “API Access”

Available Fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (Integer)</td>
<td></td>
</tr>
<tr>
<td>name (Text)</td>
<td></td>
</tr>
<tr>
<td>parentTagId (Integer)</td>
<td></td>
</tr>
<tr>
<td>ruleType</td>
<td>STATIC, GROOVY, OS_REGEX, NETWORK_RANGE, NAME_CONTAINS, INSTALLED_SOFTWARE, OPEN_PORTS, VULN_EXIST, ASSET_SEARCH, CLOUD_ASSET</td>
</tr>
<tr>
<td>color</td>
<td>Text formatted as #FFFFFF where F can be any value between 0-9 and A-F</td>
</tr>
</tbody>
</table>

Sample - Get count of all children of tag ID

API request

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
```
Qualys Asset Management & Tagging API

Tags

Request POST data

```xml
<ServiceRequest>
  <filters>
    <Criteria field="parent" operator="EQUALS">12345</Criteria>
  </filters>
</ServiceRequest>
```

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
</ServiceResponse>
```

XSD

```xml
<platform_API_server>/qps/xsd/2.0/am/tag.xsd
```
Delete Tag

/qps/rest/2.0/delete/am/tag/<id>

/qps/rest/2.0/delete/am/tag

[POST]

Delete one or more tags.

Click here for available operators

Using the NOT EQUALS operator for deleting tags could result in accidental deletion of unknown tags without any warning. To prevent accidental deletion of unknown tags, we do not support NOT EQUALS operator for delete actions.

Permissions required - Managers with full scope, other users must have these permissions: Access Permission “API Access” and Tag Permission “Delete User Tag”

Sample - Delete tag

API request


Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <SimpleTag>
      <id>12345</id>
    </SimpleTag>
  </data>
</ServiceResponse>
</data>
</ServiceResponse>
<responseCode>SUCCESS</responseCode>

XSD

<platform API server>/qps/xsd/2.0/am/tag.xsd
**Evaluate Tag**

/qps/rest/2.0/evaluate/am/tag/<id>

/qps/rest/2.0/evaluate/am/tag

[POST]

Force re-evaluation of one or more tags. Assets will be tagged only if they match the tag rule and are visible to the user. The dynamic tag evaluation feature must be turned on for your subscription in order to run this API.

Permissions required - Managers with full scope, other users must have Access Permission “API Access”

**Available Fields**

[Click here for available operators](#)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (Integer)</td>
<td></td>
</tr>
<tr>
<td>name (Text)</td>
<td></td>
</tr>
<tr>
<td>parentTagId (Integer)</td>
<td></td>
</tr>
<tr>
<td>ruleType</td>
<td>STATIC, GROOVY, OS_REGEX, NETWORK_RANGE, NAME_CONTAINS, INSTALLED_SOFTWARE, OPEN_PORTS, VULN_EXIST, ASSET_SEARCH, CLOUD_ASSET</td>
</tr>
<tr>
<td>color</td>
<td>Text formatted as #FFFFFF where F can be any value between 0-9 and A-F</td>
</tr>
</tbody>
</table>
Sample - Evaluate all tags that have Groovy Script tag rules

**API request**

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/evaluate/am/tag"
Note: “file.xml” contains the request POST data.

**Request POST data**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="ruleType" operator="EQUALS">GROOVY</Criteria>
  </filters>
</ServiceRequest>
```

**Response**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Tag>
      <id>12345</id>
      <name>Tag</name>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
      <color>#FF0000</color>
      <ruleText>asset.installedSoftwares.contains { it.name == "Windows" }</ruleText>
      <ruleType>GROOVY</ruleType>
      <children>
        <SimpleTag>
          <id>123</id>
          <name>Red</name>
        </SimpleTag>
      </children>
    </Tag>
  </data>
</ServiceResponse>
```
Qualys Asset Management & Tagging API

Tags

XSD

<platform_API_server>/qps/xsd/2.0/am/tag.xsd
List Users with their tags

/qps/rest/1.0/{action}/admin/user

Get information on users along with their tags to the authorized user. Currently, we support three actions for the users: search, count, and get details of a user.

Permissions required - Managers with full scope, other users must have Access Permission “API Access”

Search users

Search for users by using different filters for user ID, username, email, tags, and module names. If no filter is specified, all users in the user’s scope are listed.

Method: POST

XSD: user.xsd

API request

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/admin/user" < file.xml
Note: “file.xml” contains the request POST data.

Request POST data

<ServiceRequest>
    <filters>
        <Criteria field="username" operator="CONTAINS">10</Criteria>
    </filters>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
Count users

Returns the total number of users in the user's scope.
Method: POST

XSD: user.xsd

### API request

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/count/admin/user" file.xml

Note: “file.xml” contains the request POST data.

### Request POST data

```xml
<ServiceRequest>
  <filters>
    <Criteria field="username" operator="CONTAINS">10</Criteria>
  </filters>
</ServiceRequest>
```

### XML output

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

### Get user details

View details for a user in the user’s scope. You can use search action to find a user ID to use as input.

Method: GET, POST

XSD: user.xsd

### API request

curl -u "USERNAME:PASSWORD" -X GET -H "Content-type: text/xml"
XML output

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <User>
      <id>3989626</id>
      <username>user_js10</username>
      <firstName><![CDATA[John]]></firstName>
      <lastName><![CDATA[Smith]]></lastName>
      <emailAddress>john.smith@afco.com</emailAddress>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>8721654</id>
            <name><![CDATA[Unassigned Business Unit]]></name>
          </Tag>
        </list>
      </tags>
      <modules>
        <count>5</count>
        <list>
          <Module>WAS</Module>
          <Module>ADMIN</Module>
          <Module>QWEB_PCI</Module>
          <Module>ASSET_MANAGEMENT</Module>
          <Module>QWEB_VM</Module>
        </list>
      </modules>
    </User>
  </data>
</ServiceResponse>

$responseCode>SUCCESS</responseCode>
### Tag Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parentTagId</td>
<td>(integer)</td>
</tr>
<tr>
<td>color</td>
<td>(text)</td>
</tr>
<tr>
<td>ruleText</td>
<td>(text)</td>
</tr>
<tr>
<td>srcAssetGroupId</td>
<td>(integer)</td>
</tr>
<tr>
<td>srcBusinessUnitId</td>
<td>(integer)</td>
</tr>
<tr>
<td>srcOperatingSystemName</td>
<td>(text)</td>
</tr>
<tr>
<td>children</td>
<td>(TagSimpleQList)</td>
</tr>
</tbody>
</table>

**Read only fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>modified</td>
<td>(date)</td>
</tr>
</tbody>
</table>

**Associations**

TagSimpleQList - Asset tags on the associated asset. This collection to be added to and removed from is provided as a tag ID wrapped in a TagSimple element.

### TagSimple

<table>
<thead>
<tr>
<th>Name (long)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag primary key</td>
<td></td>
</tr>
<tr>
<td>name (string)</td>
<td>tag name</td>
</tr>
</tbody>
</table>
Get Tag Info

/qps/rest/2.0/get/am/tag/<id>

[GET] [POST]

Returns a single tag by ID.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the tag. Learn more

Permissions required - Managers with full scope, other users must have Access Permission “API Access”

Sample - Fetch tag

API request

```
curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/tag/12345"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Tag>
      <id>12345</id>
      <name>Test Tag</name>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
      <color>#FFFFFF</color>
      <ruleText>asset.installedSoftwares.contains { it.name == "Windows" }</ruleText>
      <ruleType>GROOVY</ruleType>
      <children>
        <list/>
    ```
Qualys Asset Management & Tagging API

Tags

```xml
</children>
</Tag>
</data>
</ServiceResponse>
<responseCode>SUCCESS</responseCode>

XSD

<platform API server>/qps/xsd/2.0/am/tag.xsd
Host Assets

Get Host Asset Info

/qps/rest/2.0/get/am/hostasset/<id>

[GET] [POST]

Returns a single host asset by ID. This API returns additional EC2 metadata of Amazon EC2 hosts when inventoried using the Qualys EC2 Connector.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the host asset. Learn more about limiting your results

Permissions required - Managers with full scope. Other users must have requested asset in their scope and these permissions: Access Permission “API Access” and Asset Management Permission “Read Asset”

Sample - Fetch host asset ID and list details

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostasset/84021"

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>False</hasMoreRecords>
  <data>
    <HostAsset>
      <id>84021</id>
      <name>10.10.23.245</name>
      <created>2018-09-12T06:21:54Z</created>
    </HostAsset>
  </data>
</ServiceResponse>
<modified>2018-09-13T01:14:34Z</modified>
<type>HOST</type>
<tags>
  <list>
    <TagSimple>
      <id>7539414</id>
      <name>Cloud Agent</name>
    </TagSimple>
  </list>
</tags>
<sourceInfo>
  <list>
    <AssetSource/>
  </list>
</sourceInfo>
<qwebHostId>18903</qwebHostId>
<os>Microsoft Windows XP Professional 5.1.2600 Service Pack 3 Build 2600</os>
<dnsHostName>XPSP2-32-27-145</dnsHostName>
<netbiosName>XPSP2-32-27-145</netbiosName>
<address>10.10.23.245</address>
<trackingMethod>QAGENT</trackingMethod>
<manufacturer>VMware, Inc.</manufacturer>
<model>VMware Virtual Platform</model>
<totalMemory>2047</totalMemory>
<timezone>-07:00</timezone>
<biosDescription>INTEL - 6040000</biosDescription>
<openPort>
  <list>
    <HostAssetOpenPort>
      <port>1900</port>
      <protocol>UDP</protocol>
    </HostAssetOpenPort>
    <HostAssetOpenPort>
      <port>7055</port>
      <protocol>TCP</protocol>
    </HostAssetOpenPort>
  </list>
</openPort>
<software>
  <list>
    <HostAssetSoftware>
      <name>Security Update for Windows XP (KB2347290)</name>
    </HostAssetSoftware>
  </list>
</software>
<version>1</version>
</HostAssetSoftware>
<HostAssetSoftware>
  <name>Security Update for Windows XP (KB950974)</name>
  <version>1</version>
</HostAssetSoftware>
</software>
<vuln>
  <list>
    <HostAssetVuln>
      <qid>118956</qid>
      <hostInstanceVulnId>296963</hostInstanceVulnId>
      <firstFound>2016-02-12T08:42:43Z</firstFound>
      <lastFound>2016-02-13T01:13:04Z</lastFound>
    </HostAssetVuln>
    <HostAssetVuln>
      <qid>119053</qid>
      <hostInstanceVulnId>296965</hostInstanceVulnId>
      <firstFound>2016-02-12T08:42:43Z</firstFound>
      <lastFound>2016-02-13T01:13:04Z</lastFound>
    </HostAssetVuln>
  </list>
</vuln>
<processor>
  <list>
    <HostAssetProcessor>
      <name>Intel Celeron processor</name>
      <speed>2799</speed>
    </HostAssetProcessor>
  </list>
</processor>
<volume>
  <list>
    <HostAssetVolume>
      <name>A:</name>
      <size>0</size>
      <free>0</free>
    </HostAssetVolume>
  </list>
</volume>
Sample - Fetch host asset ID of AWS EC2 asset and list asset details

Tags for the EC2 asset appear in the <Ec2AssetSourceSimple> element.

API request

curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostasset/709838"
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>709838</td>
</tr>
<tr>
<td>Name</td>
<td>my-ec2-target</td>
</tr>
<tr>
<td>Created</td>
<td>2017-07-27T18:14:28Z</td>
</tr>
<tr>
<td>Modified</td>
<td>2017-07-27T18:21:31Z</td>
</tr>
<tr>
<td>Type</td>
<td>HOST</td>
</tr>
<tr>
<td>Source Info</td>
<td></td>
</tr>
<tr>
<td>Asset ID</td>
<td>709838</td>
</tr>
<tr>
<td>JIRA</td>
<td>POR-6719</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:jkim@acme.com">jkim@acme.com</a></td>
</tr>
<tr>
<td>Owner</td>
<td>Jason Kim</td>
</tr>
<tr>
<td>Department</td>
<td>Security</td>
</tr>
</tbody>
</table>

XML output

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>709838</id>
      <name>my-ec2-target</name>
      <created>2017-07-27T18:14:28Z</created>
      <modified>2017-07-27T18:21:31Z</modified>
      <type>HOST</type>
      <tags>
        <list/>
      </tags>
      <sourceInfo>
        <list>
          <Ec2AssetSourceSimple>
            <firstDiscovered>2017-07-27T18:14:28Z</firstDiscovered>
            <lastUpdated>2017-07-27T19:51:03Z</lastUpdated>
            <assetId>709838</assetId>
            <ec2InstanceTags>
              <tags>
                <list>
                  <EC2Tags>
                    <key>Department</key>
                    <value>Security</value>
                  </EC2Tags>
                  <EC2Tags>
                    <key>Owner</key>
                    <value>Jason Kim</value>
                  </EC2Tags>
                  <EC2Tags>
                    <key>Email</key>
                    <value>jkim@acme.com</value>
                  </EC2Tags>
                  <EC2Tags>
                    <key>JIRA</key>
                    <value>POR-6719</value>
                  </EC2Tags>
                </list>
              </ec2InstanceTags>
            </sourceInfo>
          </Ec2AssetSourceSimple>
        </list>
      </sourceInfo>
    </HostAsset>
  </data>
</ServiceResponse>
```
<EC2Tags>
    <key>Name</key>
    <value>my-ec2-target</value>
</EC2Tags>
<EC2Tags>
    <key>Lifecycle</key>
    <value>20171231</value>
</EC2Tags>
</list>
</tags>
</ec2InstanceTags>
<availabilityZone>us-east-1e</availabilityZone>
<instanceId>i-023b166432b1c7afc</instanceId>
<instanceType>t2.medium</instanceType>
<createdDate>2017-07-27T19:58:34Z</createdDate>
<instanceState>STOPPED</instanceState>
<groupId>sg-6b619117</groupId>
<groupName>default</groupName>
<spotInstance>true</spotInstance>
<accountId>205767712438</accountId>
<subnetId>subnet-7bbcd56</subnetId>
<vpcId>vpc-2da7154b</vpcId>
<region>us-east-1</region>
<zone>VPC</zone>
<imageId>ami-22ce4934</imageId>
<publicIpAddress>127.0.0.1</publicIpAddress>
<privateIpAddress>10.97.15.117</privateIpAddress>
<monitoringEnabled>false</monitoringEnabled>
</Ec2AssetSourceSimple>
</list>
</sourceInfo>
<qwebHostId>12864</qwebHostId>
<os>Linux</os>
<address>10.97.15.117</address>
<trackingMethod>INSTANCE_ID</trackingMethod>
<openPort>
    <list/>
</openPort>
<software>
    <list/>
</software>
Qualys Asset Management & Tagging API
Host Assets

Sample - Fetch host asset ID with docker information

Tags for the docker information appear in the <dockerInfo> element.

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostasset/7727721"

XML output

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
</ServiceResponse>
<data>
  <HostAsset>
    <id>7727721</id>
    <name>10.113.198.121</name>
    <created>2018-06-15T11:51:26Z</created>
    <modified>2018-06-15T11:51:26Z</modified>
    <type>HOST</type>
    <tags>
      <list>
        <TagSimple>
          <id>8910214</id>
          <name>SSD27701</name>
        </TagSimple>
        <TagSimple>
          <id>9252992</id>
          <name>All_data1</name>
        </TagSimple>
      </list>
    </tags>
    <qwebHostId>707520</qwebHostId>
    <lastVulnScan>2018-06-15T11:48:58Z</lastVulnScan>
    <os>CentOS Linux 7.2.1511</os>
    <address>10.113.198.121</address>
    <trackingMethod>IP</trackingMethod>
    <openPort>
      <list>
        <HostAssetOpenPort>
          <port>8080</port>
          <protocol>TCP</protocol>
          <serviceId>1180</serviceId>
          <serviceName>HyperText Transport Protocol</serviceName>
        </HostAssetOpenPort>
      </list>
    </openPort>
    <vuln>
      <list>
        <HostAssetVuln>
          <qid>45038</qid>
          <hostInstanceVulnId>151189845</hostInstanceVulnId>
          <lastFound>2018-06-15T11:48:58Z</lastFound>
        </HostAssetVuln>
      </list>
    </vuln>
  </HostAsset>
</data>
Qualys Asset Management & Tagging API
Host Assets

XSD

/platform_API_server>/qps/xsd/2.0/am/hostasset.xsd
Create Host Asset

/qps/rest/2.0/create/am/hostasset

[POST]

Create one or more host assets using writable fields and collections. It is a good idea to attach tags that will make new assets visible to the current user if that user does not have permission to see all assets. Otherwise users will not be able to see or modify the new assets until an administrator or process attaches the appropriate tags to them.

What's next? After you’ve created host assets you need to activate them to make them available for scanning and reporting. Learn more on Activating Host Assets

Permissions required - Managers with full scope. Other users must have these permissions: Access Permission “API Access” and Asset Management Permission “Create Asset”

Sample - Create new host asset with tags

API request

```
```

Note: “file.xml” contains the request POST data.

Request POST data

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <HostAsset>
      <name>My Windows Asset</name>
      <os>Windows 7</os>
      <dnsHostName>localhost</dnsHostName>
      <netbiosName>TEST</netbiosName>
      <netbiosNetworkId>10</netbiosNetworkId>
    </HostAsset>
  </data>
</ServiceRequest>
```
<networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
<address>127.0.0.1</address>
<trackingMethod>IP</trackingMethod>
<tags>
  <set>
    <TagSimple><id>12345</id></TagSimple>
    <TagSimple><id>54321</id></TagSimple>
  </set>
</tags>
<software>
  <set>
    <HostAssetSoftware>
      <name>Photoshop</name>
      <version>9</version>
    </HostAssetSoftware>
  </set>
</software>
</HostAsset>
</data>
</ServiceRequest>

Response
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>2020094</id>
      <name>My Windows Asset</name>
      <created>2018-09-06T19:16:35Z</created>
      <modified>2018-09-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>12345</id>
            <name>Tag 1</name>
          </TagSimple>
          <TagSimple>
            <id>54321</id>
          </TagSimple>
        </list>
      </tags>
    </HostAsset>
  </data>
</ServiceResponse>
Sample - Bulk creation of assets

**API request**

```bash
```

Note: “file.xml” contains the request POST data.

**Request POST data**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
<data>
<HostAsset>
  <name>My Windows Asset</name>
</HostAsset>
</data>
</ServiceRequest>
```
<xml version="1.0" encoding="UTF-8"?>
<ServiceResponse

<os>Windows 8</os>
<dnsHostName>localhost13</dnsHostName>
<netbiosName>TEST</netbiosName>
<netbiosNetworkId>10</netbiosNetworkId>
<networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
<address>13.0.0.1</address>
<trackingMethod>IP</trackingMethod>
<software>
  <set>
    <HostAssetSoftware>
      <name>Photoshop</name>
      <version>9</version>
    </HostAssetSoftware>
  </set>
</software>
</HostAsset>

<HostAsset>
  <name>My Windows Asset</name>
  <os>Windows 8</os>
  <dnsHostName>localhost14</dnsHostName>
  <netbiosName>TEST</netbiosName>
  <netbiosNetworkId>10</netbiosNetworkId>
  <networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
  <address>14.0.0.1</address>
  <trackingMethod>IP</trackingMethod>
  <software>
    <set>
      <HostAssetSoftware>
        <name>Photoshop</name>
        <version>9</version>
      </HostAssetSoftware>
    </set>
  </software>
</HostAsset>
</data>
</ServiceResponse>
Qualys Asset Management & Tagging API

Host Assets

```xml
<responseCode>SUCCESS</responseCode>
<count>2</count>
<data>
  <HostAsset>
    <id>2899060</id>
    <name>My Windows Asset</name>
    <created>2016-04-01T16:57:50Z</created>
    <modified>2016-04-01T16:57:50Z</modified>
    <type>HOST</type>
    <tags>
      <list/>
    </tags>
    <sourceInfo>
      <list/>
    </sourceInfo>
    <os>Windows 8</os>
    <dnsHostName>localhost13</dnsHostName>
    <netbiosName>TEST</netbiosName>
    <netbiosNetworkId>10</netbiosNetworkId>
    <networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
    <address>13.0.0.1</address>
    <trackingMethod>IP</trackingMethod>
    <openPort>
      <list/>
    </openPort>
    <software>
      <list>
        <HostAssetSoftware>
          <name>Photoshop</name>
          <version>9</version>
        </HostAssetSoftware>
        </list>
    </software>
    <vuln>
      <list/>
    </vuln>
    <processor>
      <list/>
    </processor>
    <volume>
```
<list/>
</volume>

<account>
  <list/>
</account>

<networkInterface>
  <list>
    <HostAssetInterface>
      <hostname>localhost13</hostname>
      <type>LOCAL</type>
      <address>13.0.0.1</address>
    </HostAssetInterface>
  </list>
</networkInterface>

<HostAsset>
  <id>2899061</id>
  <name>My Windows Asset</name>
  <created>2016-04-01T16:57:51Z</created>
  <modified>2016-04-01T16:57:51Z</modified>
  <type>HOST</type>
  <tags>
    <list/>
  </tags>
  <sourceInfo>
    <list/>
  </sourceInfo>
  <os>Windows 8</os>
  <dnsHostName>localhost14</dnsHostName>
  <netbiosName>TEST</netbiosName>
  <netbiosNetworkId>10</netbiosNetworkId>
  <networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
  <address>14.0.0.1</address>
  <trackingMethod>IP</trackingMethod>
  <openPort>
    <list/>
  </openPort>
  <software>
    <list>
      <HostAssetSoftware>
        <name>Photoshop</name>
        <version>9</version>
      </HostAssetSoftware>
    </list>
  </software>
</HostAsset>
<list>
    <software>
    </software>
    <vuln>
        <list/>
    </vuln>
    <processor>
        <list/>
    </processor>
    <volume>
        <list/>
    </volume>
    <account>
        <list/>
    </account>
    <networkInterface>
        <list>
            <HostAssetInterface>
                <hostname>localhost14</hostname>
                <type>LOCAL</type>
                <address>14.0.0.1</address>
            </HostAssetInterface>
        </list>
    </networkInterface>
</HostAsset>
</data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/hostasset.xsd
Update Host Asset

/qps/rest/2.0/update/am/asset<id>

/qps/rest/2.0/update/am/asset

[POST]

Update fields for a host asset and collections of host assets.

Using the NOT EQUALS operator for updating host assets could result in accidental update of unknown hosts assets without any warning. To prevent accidental updates of unknown host assets, we do not support NOT EQUALS operator for update actions.

Permissions required - Managers with full scope, other users must have the requested assets in their scope and these permissions: Access Permission “API Access” and Asset Management Permission “Update Asset”

Sample - Update some fields for host asset ID

API request


Note: “file.xml” contains the request POST data.

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <Asset><data>
    <HostAsset>
      <name>Updated Name</name>
      <os>WINDOWS 95</os>
      <dnsHostName>win95.old.corp.net</dnsHostName>
    </HostAsset>
  </data>
</ServiceRequest>
Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>2020094</id>
      <name>Updated Name</name>
      <os>WINDOWS 95</os>
      <dnsHostName>win95.old.corp.net</dnsHostName>
      <created>2018-09-06T19:16:35Z</created>
      <modified>2018-09-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list />
      </tags>
      <sourceInfo>
        <list/>
      </sourceInfo>
      <netbiosName>TEST</netbiosName>
      <netbiosNetworkId>10</netbiosNetworkId>
      <networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
      <address>127.0.0.1</address>
      <trackingMethod>IP</trackingMethod>
      <openPort>
        <list/>
      </openPort>
      <software>
        <list/>
      </software>
      <vuln>
        <list/>
      </vuln>
    </HostAsset>
  </data>
</ServiceResponse>

Sample - Update some fields for host assets that have names containing the word OLD
## API request

```bash
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/update/am/hostasset" <
file.xml
```

Note: “file.xml” contains the request POST data.

## Request POST data

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">OLD</Criteria>
  </filters>
  <data>
    <HostAsset>
      <tags>
        <add>
          <TagSimple><id>12345</id></TagSimple>
        </add>
        <remove>
          <TagSimple><id>54321</id></TagSimple>
        </remove>
      </tags>
      <software>
        <set>
          <HostAssetSoftware>
            <name>Windows</name>
            <version>95</version>
          </HostAssetSoftware>
        </set>
      </software>
      <openPort>
        <add>
          <HostAssetOpenPort>
            <port>8080</port>
            <protocol>TCP</protocol>
          </HostAssetOpenPort>
        </add>
      </openPort>
    </HostAsset>
  </data>
</ServiceRequest>
```
Qualys Asset Management & Tagging API

Host Assets

XML output

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>2020094</id>
      <name>Updated Name</name>
      <os>WINDOWS 95</os>
      <dnsHostName>win95.old.corp.net</dnsHostName>
      <created>2014-02-06T19:16:35Z</created>
      <modified>2014-02-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>12345</id>
            <name>Simple Tag 1</name>
          </TagSimple>
        </list>
      </tags>
      <sourceInfo>
        <list/>
      </sourceInfo>
      <netbiosName>TEST</netbiosName>
      <netbiosNetworkId>10</netbiosNetworkId>
      <networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
      <address>127.0.0.1</address>
      <trackingMethod>IP</trackingMethod>
      <openPort>
        <list>
          <HostAssetOpenPort>
            <port>8080</port>
            <protocol>TCP</protocol>
          </HostAssetOpenPort>
        </list>
      </openPort>
      <software>
        <list>
          <HostAssetSoftware>
            <name>Windows</name>
          </HostAssetSoftware>
        </list>
      </software>
    </HostAsset>
  </data>
</ServiceResponse>
```
Qualys Asset Management & Tagging API
Host Assets

XSD

<platform_API_server>/qps/xsd/2.0/am/hostasset.xsd
Search Host Assets

/qps/rest/2.0/search/am/hostasset

[POST]

Returns a list of host assets matching the provided criteria. Assets are returned when they are visible to the user (i.e. in the user’s scope).

Pagination - A maximum of 100 host assets are returned by default. To customize this specify a “preferences” tag in the POST body of your request.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for each host asset. Learn more about limiting your results

Permissions required - Managers with full scope, other users must have these permissions: Access Permission “API Access” and Asset Management Permission “Read Asset”

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qwebHostId</td>
<td>(integer)</td>
</tr>
<tr>
<td>lastVulnScan</td>
<td>(date)</td>
</tr>
<tr>
<td>lastComplianceScan</td>
<td>(date)</td>
</tr>
<tr>
<td>informationGatheredUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>os</td>
<td>(text)</td>
</tr>
</tbody>
</table>
## Host Assets

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dnsHostName</td>
<td>(text)</td>
</tr>
<tr>
<td>address</td>
<td>(text)</td>
</tr>
<tr>
<td>vulnsUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>id</td>
<td>(integer)</td>
</tr>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(text)</td>
</tr>
<tr>
<td>netbiosName</td>
<td>(string)</td>
</tr>
<tr>
<td>netbiosNetworkID</td>
<td>(text)</td>
</tr>
<tr>
<td>networkGuid</td>
<td>(text)</td>
</tr>
<tr>
<td>trackingMethod</td>
<td>(keyword)</td>
</tr>
<tr>
<td></td>
<td>NONE, IP, DNSNAME, NETBIOS, INSTANCE_ID, QAGENT</td>
</tr>
<tr>
<td>port</td>
<td>(integer)</td>
</tr>
<tr>
<td>installedSoftware</td>
<td>(text)</td>
</tr>
<tr>
<td>tagName</td>
<td>(text)</td>
</tr>
<tr>
<td>tagId</td>
<td>(integer)</td>
</tr>
<tr>
<td>update</td>
<td>(date)</td>
</tr>
</tbody>
</table>

### Assets with cloud agents

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>activationKey</td>
<td>(string) Allowed operator: EQUALS</td>
</tr>
<tr>
<td>agentConfigurationName</td>
<td>(string) Allowed operators: EQUALS,</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>agentConfigurationId</td>
<td>(long)</td>
</tr>
<tr>
<td>agentVersion</td>
<td>(string)</td>
</tr>
<tr>
<td>lastCheckedIn</td>
<td>(date)</td>
</tr>
<tr>
<td>region</td>
<td>(text)</td>
</tr>
<tr>
<td>vpclId</td>
<td>(text)</td>
</tr>
<tr>
<td>imageld</td>
<td>(text)</td>
</tr>
<tr>
<td>instanceId</td>
<td>(text)</td>
</tr>
<tr>
<td>accountId</td>
<td>(text)</td>
</tr>
<tr>
<td>instanceState</td>
<td>(text)</td>
</tr>
<tr>
<td>subnetId</td>
<td>(text)</td>
</tr>
<tr>
<td>privateDnsName</td>
<td>(text)</td>
</tr>
<tr>
<td>awsTagKey</td>
<td>(text)</td>
</tr>
<tr>
<td>awsTagValue</td>
<td>(text)</td>
</tr>
</tbody>
</table>

For EC2 assets, apart from instanceState, awsTagKey, and awsTagValue, all other parameters are case sensitive. All EC2 parameters support text input with EQUALS operator. Additionally, the instanceState parameter supports EQUALS, NOT EQUALS. The awsTagKey and awsTagValue parameters support EQUALS, CONTAINS.
Sample - Search host assets

Find host assets with a Windows operating system that are tracked by Instance ID

API request

Note: “file.xml” contains the request POST data.

Request POST data

<ServiceRequest>
  <filters>
    <Criteria field="os" operator="EQUALS">Windows</Criteria>
    <Criteria field="trackingMethod" operator="EQUALS">INSTANCE_ID</Criteria>
  </filters>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>2020094</id>
      <name>Updated Name</name>
      <os>Windows</os>
      <dnsHostName>win95.old.corp.net</dnsHostName>
      <created>2018-09-06T19:16:35Z</created>
      <modified>2018-09-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list />
      </tags>
    </HostAsset>
  </data>
</ServiceResponse>
Sample - Find cloud agents with a specific agent version

**API request**

```bash
```

**Request POST data**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="agentVersion" operator="EQUALS">1.4.5.168</Criteria>
    <Criteria field="tagName" operator="EQUALS">Cloud Agent</Criteria>
  </filters>
</ServiceRequest>
```

**Response**

```
<list/>
</sourceInfo>
<netbiosName>TEST</netbiosName>
<netbiosNetworkId>10</netbiosNetworkId>
<networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
<address>127.0.0.1</address>
<trackingMethod>INSTANCE_ID</trackingMethod>
<openPort>
  <list/>
</openPort>
<software>
  <list/>
</software>
<vuln>
  <list/>
</vuln>
</HostAsset>
</data>
</ServiceResponse>
```
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>6</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <HostAsset>
      <id>3043442</id>
      <name>102115-M83</name>
      <created>2016-11-04T11:43:40Z</created>
      <modified>2016-11-08T22:35:53Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>8832525</id>
            <name>Cloud Agent</name>
          </TagSimple>
        </list>
      </tags>
      <sourceInfo>
        <list>
          <AssetSource/>
        </list>
      </sourceInfo>
      <qwebHostId>12688456922</qwebHostId>
      <dnsHostName>102115-M83</dnsHostName>
      <agentInfo>
        <agentVersion>1.4.5.168</agentVersion>
        <agentId>2e689bb2-53ab-4a58-be0a-a7576964f310</agentId>
        <status>STATUS_INACTIVE</status>
        <lastCheckedIn>2016-10-21T19:03:30Z</lastCheckedIn>
        <connectedFrom>10.100.11.163</connectedFrom>
        <chirpStatus>Manifest Downloaded</chirpStatus>
        <platform>Windows</platform>
        <agentConfiguration>
          <id>8099</id>
          <name>Initial Profile - SSN3</name>
        </agentConfiguration>
        <activationKey>
          <activationId>3ae32b8d-a8cf-4c0e-a477-
        </activationKey>
      </agentInfo>
    </HostAsset>
  </data>
</ServiceResponse>
Sample - Find host assets with specific ID containing docker information

**API request**

```
```

Note: “file.xml” contains the request POST data.
**Request POST data**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="EQUALS">7727721</Criteria>
  </filters>
</ServiceRequest>
```

**XML output**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <HostAsset>
      <id>7727721</id>
      <name>10.113.198.121</name>
      <created>2018-06-15T11:51:26Z</created>
      <modified>2018-06-15T11:51:26Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>8910214</id>
            <name>SSD27701</name>
          </TagSimple>
          <TagSimple>
            <id>9252992</id>
            <name>All_data1</name>
          </TagSimple>
        </list>
      </tags>
      <qwebHostId>707520</qwebHostId>
      <lastVulnScan>2018-06-15T11:48:58Z</lastVulnScan>
      <os>CentOS Linux 7.2.1511</os>
      <address>10.113.198.121</address>
      <trackingMethod>IP</trackingMethod>
      <openPort>
        <list/>
      </openPort>
    </HostAsset>
  </data>
</ServiceResponse>
```
<port>8080</port>
<protocol>TCP</protocol>
<serviceId>1180</serviceId>
<serviceName>HyperText Transport Protocol</serviceName>
</HostAssetOpenPort>

</list>
</openPort>
<vuln>
<list>
<HostAssetVuln>
<qid>6</qid>
<hostInstanceVulnId>151189838</hostInstanceVulnId>
<lastFound>2018-06-15T11:48:58Z</lastFound>
</HostAssetVuln>
<HostAssetVuln>
<qid>45038</qid>
<hostInstanceVulnId>151189845</hostInstanceVulnId>
<lastFound>2018-06-15T11:48:58Z</lastFound>
</HostAssetVuln>
</list>
</vuln>

<networkInterface>
<list>
<HostAssetInterface>
<type>LOCAL</type>
<address>10.113.198.121</address>
</HostAssetInterface>
</list>
</networkInterface>
</ServiceResponse>
Sample - Find host assets with specific ID containing split manifest version information for VM, PC, or SCA

**API request**

```
```

Note: “file.xml” contains the request POST data.

**Request POST data**

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="EQUALS">7866685</Criteria>
  </filters>
</ServiceRequest>
```

**XML output**

```
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <HostAsset>
      <id>7866685</id>
      <name>ip-172-31-3-82.ap-south-1.compute.internal</name>
      <created>2018-08-01T09:34:44Z</created>
      <modified>2018-08-10T08:39:49Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>10125654</id>
            <name>Cloud Agent</name>
          </TagSimple>
        </list>
      </tags>
    </HostAsset>
  </data>
</ServiceResponse>
```
<sourceInfo>
  <list>
    <AssetSource/>
    <Ec2AssetSourceSimple>
      <assetId>7866685</assetId>
      <type>EC_2</type>
      <firstDiscovered>2018-08-01T09:34:45Z</firstDiscovered>
      <lastUpdated>2018-08-01T09:34:45Z</lastUpdated>
      <reservationId>r-0cd44450f874d4a08</reservationId>
      <availabilityZone>ap-south-1b</availabilityZone>
      <privateDnsName>ip-172-31-3-82.ap-south-1.compute.internal</privateDnsName>
      <publicDnsName>ec2-13-232-170-59.ap-south-1.compute.amazonaws.com</publicDnsName>
      <localHostname>ip-172-31-3-82.ap-south-1.compute.internal</localHostname>
      <instanceId>i-0ce729520a8a7d696</instanceId>
      <instanceType>t2.micro</instanceType>
      <instanceState>RUNNING</instanceState>
      <groupId>sg-608b270a</groupId>
      <groupName>launch-wizard-4</groupName>
      <spotInstance>false</spotInstance>
      <accountId>383031258652</accountId>
      <subnetId>subnet-5a0d6a17</subnetId>
      <vpcId>vpc-39ccea50</vpcId>
      <region>ap-south-1</region>
      <zone>VPC</zone>
      <imageId>ami-5b673c34</imageId>
      <publicIpAddress>13.232.170.59</publicIpAddress>
      <privateIpAddress>172.31.3.82</privateIpAddress>
      <macAddress>0a:da:e8:58:09:fe</macAddress>
      <monitoringEnabled>false</monitoringEnabled>
    </Ec2AssetSourceSimple>
  </list>
</sourceInfo>
<lastVulnScan>2018-08-10T04:55:06Z</lastVulnScan>
<lastSystemBoot>2018-08-01T09:23:42Z</lastSystemBoot>
<lastLoggedOnUser>ec2-user</lastLoggedOnUser>
<os>Red Hat Enterprise Linux Server 7.5</os>
<dnsHostName>ip-172-31-3-82.ap-south-1.compute.internal</dnsHostName>

<agentInfo>
  <agentVersion>1.7.1.38</agentVersion>
  <agentId>66fb864e-9609-4324-8eec-48ab6cb7f260</agentId>
  <status>STATUS_ACTIVE</status>
  <lastCheckedIn>2018-08-10T08:39:42Z</lastCheckedIn>
  <connectedFrom>13.232.170.59</connectedFrom>
  <location>Mumbai,Maharashtra India</location>
  <locationGeoLatitude>18.975</locationGeoLatitude>
  <locationGeoLongitude>72.8258</locationGeoLongitude>
  <chirpStatus>Inventory Scan Complete</chirpStatus>
  <platform>Linux</platform>
  <activatedModule>AGENT_VM</activatedModule>
  <manifestVersion>
    <vm>VULNSIGS-VM-0.12.1.0-17</vm>
    <pc>VULNSIGS-PC-0.17.0.0-27</pc>
  </manifestVersion>
  <agentConfiguration>
    <id>514001</id>
    <name>My Default</name>
  </agentConfiguration>
  <activationKey>
    <activationId>f9391862-de71-4106-9478-ca14042980dd</activationId>
    <title>AWS</title>
  </activationKey>
</agentInfo>

<networkGuid>6b48277c-0742-61c1-82bb-cac0f9c4094a</networkGuid>
<address>13.232.170.59</address>
<trackingMethod>QAGENT</trackingMethod>
<totalMemory>990</totalMemory>
<timezone>UTC</timezone>

<openPort>
  <list>
    <HostAssetOpenPort>
      <port>323</port>
      <protocol>UDP</protocol>
    </HostAssetOpenPort>
  </list>
</openPort>
<volume>
  <list>
    <HostAssetVolume>
      <name>/</name>
      <size>10724814848</size>
      <free>9259859968</free>
    </HostAssetVolume>
    ...
  </list>
</volume>

<account>
  <list>
    <HostAssetAccount>
      <username>root</username>
    </HostAssetAccount>
    <HostAssetAccount>
      <username>ec2-user</username>
    </HostAssetAccount>
  </list>
</account>

<networkInterface>
  <list>
    <HostAssetInterface>
      <interfaceName>eth0</interfaceName>
      <macAddress>0a:da:e8:58:09:fe</macAddress>
      <type>LOCAL</type>
      <address>fe80:0:0:0:8da:e8ff:fe58:9fe</address>
      <gatewayAddress>172.31.0.1</gatewayAddress>
    </HostAssetInterface>
    ...
  </list>
</networkInterface>

</HostAsset>
</data>
</ServiceResponse>

Sample - Search EC2 assets in your account

API request

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/hostasset" <
file.xml
Note: "file.xml" contains the request POST data.

Request POST data
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="region" operator="EQUALS">ap-northeast-1</Criteria>
    <Criteria field="vpcId" operator="EQUALS">vpc-98a11ffd</Criteria>
    <Criteria field="accountId" operator="EQUALS">205767712438</Criteria>
    <Criteria field="privateDnsName" operator="EQUALS">ip-172-30-1-133.ap-northeast-1.compute.internal</Criteria>
  </filters>
</ServiceRequest>

XML output
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <HostAsset>
      <id>1553126</id>
      <name>ip-172-30-1-133</name>
      <created>2018-12-03T09:10:18Z</created>
      <modified>2018-12-08T10:14:40Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>7977614</id>
            <name>Cloud Agent</name>
          </TagSimple>
        </list>
      </tags>
    </HostAsset>
  </data>
</ServiceResponse>
<sourceInfo>
  <list>
    <AssetSource/>
    <Ec2AssetSourceSimple>
      <assetId>1553126</assetId>
      <type>EC_2</type>
      <firstDiscovered>2018-12-03T09:10:18Z</firstDiscovered>
      <lastUpdated>2018-12-03T09:10:18Z</lastUpdated>
      <reservationId>r-08a2a6ee33b3acd9f</reservationId>
      <availabilityZone>ap-northeast-1b</availabilityZone>
      <privateDnsName>ip-172-30-1-133.ap-northeast-1.compute.internal</privateDnsName>
      <localHostname>ip-172-30-1-133.ap-northeast-1.compute.internal</localHostname>
      <instanceId>i-07081d0a8ab051d80</instanceId>
      <instanceType>t2.micro</instanceType>
      <instanceState>RUNNING</instanceState>
      <groupId>sg-9a08a0e3</groupId>
      <groupName>launch-wizard-12</groupName>
      <accountId>205767712438</accountId>
      <subnetId>subnet-5c198e2b</subnetId>
      <vpcId>vpc-98a11ff8</vpcId>
      <region>ap-northeast-1</region>
      <zone>VPC</zone>
      <imageId>ami-92df37ed</imageId>
      <publicIpAddress>13.113.179.242</publicIpAddress>
      <privateIpAddress>172.30.1.133</privateIpAddress>
      <macAddress>06:c2:ed:39:19:98</macAddress>
    </Ec2AssetSourceSimple>
  </list>
</sourceInfo>

<qwebHostId>294355</qwebHostId>
<lastComplianceScan>2018-12-08T01:45:34Z</lastComplianceScan>
<lastVulnScan>2018-12-08T07:14:58Z</lastVulnScan>
<lastSystemBoot>2018-05-25T06:06:35Z</lastSystemBoot>
<lastLoggedOnUser>ec2-user</lastLoggedOnUser>
<os>Amazon Linux 2018.03</os>
</dnsHostName>
    </agentInfo>
    <agentInfo>
        <agentVersion>2.3.0.20</agentVersion>
        <agentId>f6e1a6be-a99a-4d79-a5b1-f339aeaf8095</agentId>
        <status>STATUS_INACTIVE</status>
        <lastCheckedIn>2018-12-08T07:15:20Z</lastCheckedIn>
        <connectedFrom>13.113.179.242</connectedFrom>
        <location>Tokyo, Tokyo Japan</location>
        <locationGeoLatitude>35.685</locationGeoLatitude>
        <locationGeoLongitude>139.7514</locationGeoLongitude>
    </agentInfo>

    <chirpStatus>Inventory Scan Complete</chirpStatus>
    <platform>Linux</platform>
    <activatedModule>AGENT_VM,AGENT_PC,FIM</activatedModule>

    <manifestVersion>
        <vm>VULNSIGS-VM-0.19.0.0-34</vm>
        <pc>VULNSIGS-PC-0.19.0.0-34</pc>
    </manifestVersion>
    <agentConfiguration>
        <id>166800</id>
        <name>27-March</name>
    </agentConfiguration>
    <activationKey>
        <activationId>8d988825-5685-4dcf-8d14-0fde25eab037</activationId>
        <title>september-2018</title>
    </activationKey>

    <networkGuid>6b48277c-0742-61c1-82bb-cac0f9c4094a</networkGuid>
    <address>13.113.179.242</address>
    <trackingMethod>QAGENT</trackingMethod>
    <totalMemory>987</totalMemory>
    <timezone>UTC</timezone>

    <openPort>
        <list>
            <HostAssetOpenPort>
                <port>57091</port>
                <protocol>UDP</protocol>
            </HostAssetOpenPort>
            ...
        </list>
    </openPort>
<openPort/>
<software>
  <list>
    <HostAssetSoftware>
      <name>acl</name>
      <version>2.2.49-6.11.amzn1</version>
    </HostAssetSoftware>
    ...
  </list>
</software>
<vuln>
  <list>
    <HostAssetVuln>
      <qid>38582</qid>
      <hostInstanceVulnId>88353071</hostInstanceVulnId>
      <firstFound>2018-12-03T22:07:32Z</firstFound>
      <lastFound>2018-12-08T07:14:58Z</lastFound>
    </HostAssetVuln>
    ...
  </list>
</vuln>
<processor>
  <list>
    <HostAssetProcessor>
      <name>Intel(R) Xeon(R)</name>
      <speed>2400</speed>
    </HostAssetProcessor>
  </list>
</processor>
<volume>
  <list>
    <HostAssetVolume>
      <name>/dev</name>
      <size>506937344</size>
      <free>506880000</free>
    </HostAssetVolume>
    ...
  </list>
</volume>
<account>
  <list>
    <HostAssetAccount>
      <username>root</username>
    </HostAssetAccount>
    ...
  </list>
</account>
<HostAssetAccount><username>ec2-user</username></HostAssetAccount></account><networkInterface><list><HostAssetInterface><hostname>ip-172-30-1-133</hostname><interfaceName>eth0      Link encap</interfaceName><macAddress>06:C2:ED:39:19:98</macAddress><type>LOCAL</type><address>172.30.1.133</address><gatewayAddress>172.30.1.1</gatewayAddress></HostAssetInterface>...
</list></networkInterface><isDockerHost>false</isDockerHost></HostAsset></data></ServiceResponse>

XSD

<platform API server>/qps/xsd/2.0/am/hostasset.xsd
Count Host Assets

/qps/rest/2.0/count/am/hostasset

[POST]

Returns the number of host assets that match the provided criteria. A host asset is counted when the asset is visible to the user (i.e. it is in the user’s scope).

Permissions required - Managers with full scope. Other users must have these permissions: Access Permission “API Access” and Asset Management Permission “Read Asset”

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qwebHostId</td>
<td>(integer)</td>
</tr>
<tr>
<td>lastVulnScan</td>
<td>(date)</td>
</tr>
<tr>
<td>lastComplianceScan</td>
<td>(date)</td>
</tr>
<tr>
<td>informationGatheredUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>os</td>
<td>(text)</td>
</tr>
<tr>
<td>dnsHostName</td>
<td>(text)</td>
</tr>
<tr>
<td>address</td>
<td>(text)</td>
</tr>
<tr>
<td>vulnsUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>id</td>
<td>(integer)</td>
</tr>
</tbody>
</table>
### Qualys Asset Management & Tagging API

#### Host Assets

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(text)</td>
</tr>
<tr>
<td>netbiosName</td>
<td>(string)</td>
</tr>
<tr>
<td>netbiosNetworkID</td>
<td>(text)</td>
</tr>
<tr>
<td>networdGuid</td>
<td>(text)</td>
</tr>
<tr>
<td>trackingMethod</td>
<td>(keyword) NONE, IP, DNSNAME, NETBIOS, INSTANCE_ID, QAGENT</td>
</tr>
<tr>
<td>port</td>
<td>(integer)</td>
</tr>
<tr>
<td>installedSoftware</td>
<td>(text)</td>
</tr>
<tr>
<td>tagName</td>
<td>(text)</td>
</tr>
<tr>
<td>tagId</td>
<td>(integer)</td>
</tr>
<tr>
<td>update</td>
<td>(date)</td>
</tr>
</tbody>
</table>

### Sample - Count host assets

#### API request

```bash
curl -u "USERNAME:PASSWORD" -X POST --data-binary @$- https://qualysapi.qualys.com/qps/rest/2.0/count/am/hostasset -H vContent-Type: application/xml" < file.xml
```

Note: “file.xml” contains the request POST data.

#### Request POST data

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
```
<filters>
  <Criteria field="os" operator="EQUALS">Windows</Criteria>
</filters>

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>235</count>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/hostasset.xsd
Delete Host Asset

/qps/rest/2.0/delete/am/hostasset/<id>

/qps/rest/2.0/delete/am/hostasset

[POST]

Delete one or more host assets.

Using the NOT EQUALS operator for deleting host assets could result in accidental deletion of unknown host assets without any warning. To prevent accidental deletion of unknown host assets, we do not support NOT EQUALS operator for delete actions.

Permissions required - Managers with full scope. Other users must have these permissions: Access Permission “API Access” and Asset Management Permission “Delete Asset”

Sample - Delete host assets with the tag “To Delete”

API request

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/delete/am/hostasset" < file.xml
Note: “file.xml” contains the request POST data.

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">To Delete</Criteria>
  </filters>
</ServiceRequest>

Response
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HostAsset>
            <id>2020094</id>
        </HostAsset>
    </data>
</ServiceResponse>

XSD

<platform API server>/qps/xsd/2.0/am/hostasset.xsd
Activate Host Asset

/qps/rest/2.0/activate/am/hostasset/<id>?module=QWEB_VM
/qps/rest/2.0/activate/am/hostasset?module=QWEB_VM
/qps/rest/2.0/activate/am/hostasset/<id>?module=QWEB_PC
/qps/rest/2.0/activate/am/hostasset?module=QWEB_PC

[POST]

Activate one or more assets to make them available in your account for scanning and reporting. You’ll want to activate newly created hosts to make them available in the Vulnerability Management (VM) module and/or the Policy Compliance (PC) module.

Permissions required - Users with full scope. Other users must have requested assets in their scope and Access Permission “API Access”.

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qwebHostId</td>
<td>(integer)</td>
</tr>
<tr>
<td>lastVulnScan</td>
<td>(date)</td>
</tr>
<tr>
<td>lastComplianceScan</td>
<td>(date)</td>
</tr>
<tr>
<td>informationGatheredUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>os</td>
<td>(text)</td>
</tr>
<tr>
<td>dnsHostName</td>
<td>(text)</td>
</tr>
</tbody>
</table>
## Qualys Asset Management & Tagging API

### Host Assets

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>(text)</td>
</tr>
<tr>
<td>vulnsUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>id</td>
<td>(integer)</td>
</tr>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(text)</td>
</tr>
<tr>
<td>netbiosName</td>
<td>(string)</td>
</tr>
<tr>
<td>netbiosNetworkID</td>
<td>(text)</td>
</tr>
<tr>
<td>networdGuid</td>
<td>(text)</td>
</tr>
<tr>
<td>trackingMethod</td>
<td>(keyword)</td>
</tr>
<tr>
<td>port</td>
<td>(integer)</td>
</tr>
<tr>
<td>installedSoftware</td>
<td>(text)</td>
</tr>
<tr>
<td>tagName</td>
<td>(text)</td>
</tr>
<tr>
<td>tagId</td>
<td>(integer)</td>
</tr>
<tr>
<td>update</td>
<td>(date)</td>
</tr>
</tbody>
</table>

### Sample - Activate host assets with tag “Export to VM”

#### API request

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
```
Qualys Asset Management & Tagging API

Host Assets

"https://qualysapi.qualys.com/qps/rest/2.0/activate/am/hostasset" <
file.xml
Note: “file.xml” contains the request POST data.

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">Export to VM</Criteria>
  </filters>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8" ?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>2020094</id>
      <name>Updated Name</name>
      <os>WINDOWS 95</os>
      <dnsHostName>win95.old.corp.net</dnsHostName>
      <created>2018-09-06T19:16:35Z</created>
      <modified>2018-09-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list />
      </tags>
      <sourceInfo>
        <list/>
      </sourceInfo>
      <netbiosName>TEST</netbiosName>
      <netbiosNetworkId>10</netbiosNetworkId>
      <networkGuid>66bf43c8-7392-4257-b856-a320fde231eb</networkGuid>
      <address>127.0.0.1</address>
      <trackingMethod>IP</trackingMethod>
      <openPort>
        <list/>
      </openPort>
    </HostAsset>
  </data>
</ServiceResponse>
Qualys Asset Management & Tagging API

Host Assets

```
</openPort>
<software>
  <list/>
</software>
<vuln>
  <list/>
</vuln>
</HostAsset>
</data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/hostasset.xsd
## Host Asset Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>os</td>
<td>(string)</td>
</tr>
<tr>
<td>dnsHostName</td>
<td>(string)</td>
</tr>
<tr>
<td>netbiosName</td>
<td>(string)</td>
</tr>
<tr>
<td>netbiosNetworkId</td>
<td>(integer)</td>
</tr>
<tr>
<td>networkGuid</td>
<td>(uuid)</td>
</tr>
<tr>
<td>address</td>
<td>(string)</td>
</tr>
<tr>
<td>trackingMethod</td>
<td>(AssetTrackingMethod: NONE, IP, DNSNAME, NETBIOS, INSTANCE_ID, QAGENT)</td>
</tr>
<tr>
<td>openPort</td>
<td>(HostAssetOpenPortQList)</td>
</tr>
<tr>
<td>software</td>
<td>(HostAssetSoftwareQList)</td>
</tr>
</tbody>
</table>

**Read only fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qwebHostId</td>
<td>(long)</td>
</tr>
<tr>
<td>lastVulnScan</td>
<td>(date)</td>
</tr>
<tr>
<td>lastComplianceScan</td>
<td>(date)</td>
</tr>
<tr>
<td>vulnsUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>informationGatheredUpdated</td>
<td>(date)</td>
</tr>
<tr>
<td>account</td>
<td>(HostAssetAccount)</td>
</tr>
<tr>
<td>biosDescription</td>
<td>(string)</td>
</tr>
<tr>
<td>manufacturer</td>
<td>(string)</td>
</tr>
<tr>
<td>model</td>
<td>(string)</td>
</tr>
<tr>
<td>networkInterface</td>
<td>(HostAssetInterface)</td>
</tr>
<tr>
<td>processor</td>
<td>(HostAssetProcessor)</td>
</tr>
<tr>
<td>timezone</td>
<td>(string)</td>
</tr>
<tr>
<td>totalMemory</td>
<td>(long)</td>
</tr>
<tr>
<td>volume</td>
<td>(HostAssetVolume)</td>
</tr>
</tbody>
</table>

**EC2 fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>region</td>
<td>(text)</td>
</tr>
<tr>
<td>vpclid</td>
<td>(text)</td>
</tr>
<tr>
<td>imageld</td>
<td>(text)</td>
</tr>
<tr>
<td>instanceldId</td>
<td>(text)</td>
</tr>
<tr>
<td>accountId</td>
<td>(text)</td>
</tr>
<tr>
<td>instanceState</td>
<td>(text) PENDING, RUNNING, TERMINATED, STOPPED, etc.</td>
</tr>
<tr>
<td>subnetId</td>
<td>(text)</td>
</tr>
<tr>
<td>privateDnsName</td>
<td>(text)</td>
</tr>
</tbody>
</table>
awsTagKey (text)

awsTagValue (text)

Associations

HostAssetOpenPortQList - Open ports (HostAssetOpenPortList) detected or explicitly added to the asset. This collection is keyed off of the port and protocol.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>integer</td>
</tr>
<tr>
<td>protocol</td>
<td>protocol (TCP, UDP, ICMP)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceId</td>
<td>integer</td>
</tr>
<tr>
<td>serverName</td>
<td>string (name of the service detected on the port - read only)</td>
</tr>
</tbody>
</table>

HostAssetSoftwareQList - A list of software (HostAssetSoftware) installed on the machine, keyed on the name.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
</tr>
<tr>
<td>version</td>
<td>string</td>
</tr>
</tbody>
</table>
HostAssetVulnQList - A list of vulnerabilities detected on the host. Only vulnerabilities flagged as found will be returned. More detailed information about each detected vulnerability can be obtained from the HostInstanceVuln resource, cross referenced by the hostInstanceVulnId field. The HostInstanceVuln can also be used to find previously detected vulnerabilities that are currently marked as not found.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>qid</td>
<td>long</td>
</tr>
<tr>
<td>hostInstanceVulnID</td>
<td>long</td>
</tr>
<tr>
<td>firstFound</td>
<td>date</td>
</tr>
<tr>
<td>lastFound</td>
<td>date</td>
</tr>
</tbody>
</table>
Assets

Get Asset Info

/qps/rest/2.0/get/am/asset/<id>

[GET] [POST]

Returns a single asset by ID.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the host asset. Learn more about limiting your results

Permissions required - Managers with full scope. Other users must have requested asset in their scope and these permissions: Access Permission “API Access” and Asset Management Permission “Read Asset”

Sample - Fetch asset ID and list details

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/asset/12345"

Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>12345</id>
      <name>My Windows Asset</name>
      <created>2014-02-06T19:16:35Z</created>
      <modified>2014-02-06T19:16:35Z</modified>
      <type>HOST</type>
    </Asset>
  </data>
</ServiceResponse>
Qualys Asset Management & Tagging API

Assets

XSD

<platform API server>/qps/xsd/2.0/am/asset.xsd
Update Asset

/qps/rest/2.0/update/am/asset<id>
/qps/rest/2.0/update/am/asset

[POST]

Update fields for an asset and collections of assets. Only the name and tags can be modified.

Using the NOT EQUALS operator for updating assets could result in accidental update of unknown assets without any warning. To prevent accidental updates of unknown assets, we do not support NOT EQUALS operator for update actions.

Permissions required - Managers with full scope, other users must have the requested assets in their scope and these permissions: Access Permission “API Access” and Asset Management Permission “Update Asset”

Sample - Update tag and give it another name

API request

Note: “file.xml” contains the request POST data.

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <Asset>
      <name>Updated Name</name>
    </Asset>
  </data>
</ServiceRequest>
Qualys Asset Management & Tagging API

Assets

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>12345</id>
      <name>Updated Name</name>
      <created>2014-02-06T19:16:35Z</created>
      <modified>2014-02-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>12345</id>
            <name>Tag 1</name>
          </TagSimple>
          <TagSimple>
            <id>54321</id>
            <name>Tag 2</name>
          </TagSimple>
        </list>
      </tags>
    </Asset>
  </data>
</ServiceResponse>

Sample - Update tags that have tag names containing the word DELETED

API request

Note: “file.xml” contains the request POST data.

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">DELETED</Criteria>
  </filters>
  <data>
    <Asset>
      <tags>
        <add>
          <TagSimple><id>12345</id></TagSimple>
        </add>
      </tags>
    </Asset>
  </data>
</ServiceRequest>

XML output

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>543</id>
      <name>Old Asset (DELETED)</name>
      <created>2014-02-06T19:16:35Z</created>
      <modified>2014-02-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>12345</id>
            <name>Tag 1</name>
          </TagSimple>
        </list>
      </tags>
    </Asset>
  </data>
</ServiceResponse>

XSD
<platform_API_server>/qps/xsd/2.0/am/asset.xsd
Search Assets

/qps/rest/2.0/search/am/asset

[POST]

Returns a list of assets matching the provided criteria. Assets are returned when they are visible to the user (i.e. in the user’s scope).

Pagination - A maximum of 100 host assets are returned by default. To customize this specify a “preferences” tag in the POST body of your request.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for each host asset. Learn more about limiting your results

Permissions required - Managers with full scope, other users must have these permissions: Access Permission “API Access” and Asset Management Permission “Read Asset”

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer)</td>
</tr>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>updated</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN</td>
</tr>
</tbody>
</table>
Sample - Find an asset with a particular tag

API request

```bash
```

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">To Delete</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>543</id>
      <name>Old Asset (To Delete)</name>
      <created>2014-02-06T19:16:35Z</created>
      <modified>2014-02-06T19:16:35Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>12345</id>
            <name>Tag 1</name>
          </TagSimple>
        </list>
      </tags>
    </Asset>
  </data>
</ServiceResponse>
```
Qualys Asset Management & Tagging API

Assets

XSD

<platform_API_server>/qps/xsd/2.0/am/asset.xsd
Count Assets

/qps/rest/2.0/count/am/asset

[POST]

Returns the number of assets that match the provided criteria. A host asset is counted when the asset is visible to the user (i.e. in the user’s scope).

Permissions required - Managers with full scope. Other users must have these permissions: Access Permission “API Access” and Asset Management Permission “Read Asset”

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer)</td>
</tr>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>updated</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN</td>
</tr>
<tr>
<td>tagName</td>
<td>(text) Parent tags of the tag will also match</td>
</tr>
<tr>
<td>tagId</td>
<td>(text) Parent tags of the tag will also match</td>
</tr>
</tbody>
</table>

Sample - Count assets with tag name "To Delete"
Qualys Asset Management & Tagging API

Assets

API request

curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/2.0/count/am/asset" < file.xml
Note: “file.xml” contains the request POST data.

Request POST data

<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">To Delete</Criteria>
  </filters>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/asset.xsd
Delete Asset

/qps/rest/2.0/delete/am/asset/<id>
/qps/rest/2.0/delete/am/asset
[POST]

Delete one or more assets.

Using the NOT EQUALS operator for deleting assets could result in accidental
deletion of unknown assets without any warning. To prevent accidental
deletion of unknown assets, we do not support NOT EQUALS operator for
delete actions.

Permissions required - Managers with full scope. Other users must have these
permissions: Access Permission “API Access” and Asset Management
Permission “Delete Asset”

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer)</td>
</tr>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>updated</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN</td>
</tr>
<tr>
<td>tagName</td>
<td>(text) Parent tags of the tag will also match</td>
</tr>
</tbody>
</table>
Sample - Delete assets with a particular tag name

**API request**

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/delete/am/asset" < file.xml
```

Note: “file.xml” contains the request POST data.

**Request POST data**

```xml
<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">To Delete</Criteria>
  </filters>
</ServiceRequest>
```

**Response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>1972521</id>
    </Asset>
  </data>
</ServiceResponse>
```

**XSD**

```xml
<platform_API_server>/qps/xsd/2.0/am/asset.xsd
```
Activate Asset

?qps/rest/2.0/activate/am/hostasset/<id>?module=QWEB_VM
?qps/rest/2.0/activate/am/hostasset?module=QWEB_VM
?qps/rest/2.0/activate/am/hostasset/<id>?module=QWEB_PC
?qps/rest/2.0/activate/am/hostasset?module=QWEB_PC

[POST]

Activate one or more assets to make them available in your account for scanning and reporting. You'll want to activate newly created hosts to make them available in the Vulnerability Management (VM) module and/or the Policy Compliance (PC) module.

Permissions required - Users with full scope. Other users must have requested assets in their scope and Access Permission “API Access”.

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer)</td>
</tr>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>updated</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN</td>
</tr>
</tbody>
</table>
tagName  (text) Parent tags of the tag will also match

Sample - Activate assets with the tag “Export to VM”

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/activate/am/asset?module=QWEB_VM" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">Export to VM</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>1972521</id>
      <name>Test Asset</name>
      <created>2013-12-11T05:12:45Z</created>
      <modified>2014-02-04T23:55:54Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <SimpleTag>
<id>12345</id>
  <name>Export to VM</name>
</SimpleTag>

<list/>
</tags>
<sourceInfo>
  <list>
    <Ec2AssetSourceSimple>
      <firstDiscovered>2014-02-06T19:14:50Z</firstDiscovered>
      <lastUpdated>2014-02-06T19:14:50Z</lastUpdated>
      <assetId>1972521</assetId>
      <availabilityZone>us-east</availabilityZone>
      <privateDnsName>ip-10-90-0-73.qualys.com</privateDnsName>
      <instanceId>i-8b545eef</instanceId>
      <instanceType>t1.micro</instanceType>
      <imageId>ami-03ad6e6a</imageId>
      <privateIpAddress>127.0.0.1</privateIpAddress>
      <monitoringEnabled>false</monitoringEnabled>
    </Ec2AssetSourceSimple>
  </list>
</sourceInfo>
<openPort>
  <list/>
</openPort>
<software>
  <list/>
</software>
<vuln>
  <list/>
</vuln>
</Asset>
</data>
</ServiceResponse>

XSD
<platform_API_server>/qps/xsd/2.0/am/asset.xsd
Asset Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text)</td>
</tr>
<tr>
<td>tags</td>
<td>(string)</td>
</tr>
<tr>
<td>id</td>
<td>(long)</td>
</tr>
<tr>
<td>created</td>
<td>(date)</td>
</tr>
<tr>
<td>modified</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN</td>
</tr>
<tr>
<td>sourceInfo</td>
<td>(AssetSourceQList)</td>
</tr>
</tbody>
</table>

Associations

TagSimpleQList - Asset tags on the associated asset. This collection to be added to and removed from by providing a tag ID wrapped in a TagSimple element.

<table>
<thead>
<tr>
<th>TagSimple</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>name</td>
</tr>
</tbody>
</table>

AssetSourceQList - Source information for the associated asset. At the moment this is used exclusively for assets that are in Amazon EC2 but may
contain additional types in the future. As such, elements will always be of type `Ec2AssetSourceSimple`.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>availabilityZone</td>
<td>(string)</td>
</tr>
<tr>
<td>privateDnsName</td>
<td>(string)</td>
</tr>
<tr>
<td>publicDnsName</td>
<td>(string)</td>
</tr>
<tr>
<td>instanceID</td>
<td>(string)</td>
</tr>
<tr>
<td>instanceType</td>
<td>(string)</td>
</tr>
<tr>
<td>imageID</td>
<td>(string)</td>
</tr>
<tr>
<td>publicIpAddress</td>
<td>(string)</td>
</tr>
<tr>
<td>privateIpAddress</td>
<td>(string)</td>
</tr>
<tr>
<td>monitoringEnabled</td>
<td>(boolean)</td>
</tr>
<tr>
<td>instanceState</td>
<td>(AssetSourceStateCode: PENDING, RUNNING, SHUTTING_DOWN, TERMINATED, STOPPING, STOPPED, UNSUPPORTED)</td>
</tr>
</tbody>
</table>
Host Instance Vulnerability data

Get Vulnerability Info

/qps/rest/2.0/get/am/hostinstancevuln/<id>

[GET]

Returns a single host instance vulnerability data by ID.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the host instance vulnerability. Learn more about limiting your results

Permissions required - Managers with Full Scope. Users without Full Scope must have these account settings: 1) scope includes the requested asset, and 2) permissions include: Access Permission “API Access” and Asset Management Permission “Read Asset”.

Sample - Fetch a host instance vulnerability

API request

```
curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostinstancevuln/12345"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HostInstanceVuln>
            <id>9534081</id>
            <hostAssetId>1543621</hostAssetId>
        </HostInstanceVuln>
    </data>
</ServiceResponse>
```
<qid>38167</qid>
<port>25</port>
<ssl>true</ssl>
<found>true</found>
<ignored>false</ignored>
<disabled>false</disabled>
<updated>2012-10-19T21:56:23Z</updated>
<protocol>TCP</protocol>
<source>HOST</source>
</HostInstanceVuln>
</data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/hostinstancevuln.xsd
Search Vulnerabilities

/qps/rest/2.0/search/am/hostinstancevuln

[POST]

Returns a list of host instance vulnerabilities that match the provided criteria. These vulnerabilities are returned when the hosts are visible to the user (i.e. in the user's scope).

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for each vulnerability. Learn more about limiting your results.

Pagination - A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.

Permissions required - Managers with Full Scope. Users without Full Scope must have these account settings: 1) scope includes the requested asset, and 2) permissions include: Access Permission “API Access” and Asset Management Permission “Read Asset”.

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(long) The primary host instance vulnerability key.</td>
</tr>
<tr>
<td>hostAssetId</td>
<td>(long) The ID of the host asset where the vulnerability was found.</td>
</tr>
<tr>
<td>created</td>
<td>(date) The date the vulnerability was added to the KnowledgeBase.</td>
</tr>
<tr>
<td>found</td>
<td>(boolean) Set to true if the QID was detected on the host by the latest scan of that host.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>firstFound</td>
<td>(date) The date/time the vulnerability was first detected on the host.</td>
</tr>
<tr>
<td>lastfound</td>
<td>(date) The most recent date/time the vulnerability was detected on the host.</td>
</tr>
<tr>
<td>lastScanned</td>
<td>(date) The most recent date/time the vulnerability was tested for the host.</td>
</tr>
<tr>
<td>qid</td>
<td>(long) The Qualys vulnerability ID of the vulnerability.</td>
</tr>
<tr>
<td>disabled</td>
<td>(boolean) Set to true if the QID is marked as disabled in your subscription. Set to false if the QID is not marked disabled.</td>
</tr>
<tr>
<td>fqdn</td>
<td>(string) The fully qualified domain name of the host.</td>
</tr>
<tr>
<td>ssl</td>
<td>(boolean) Set to true if the vulnerability was detected over SSL. Set to false if the vulnerability was not detected over SSL. This element is not returned for information gathered.</td>
</tr>
<tr>
<td>updated</td>
<td>(date) The last date/time the vulnerability data was updated for the host.</td>
</tr>
<tr>
<td>ignored</td>
<td>(boolean) Set to true if the QID/host/port is marked as ignored in your subscription. Set to false if the QID/host/port is not marked ignored.</td>
</tr>
<tr>
<td>protocol</td>
<td>(string) The protocol the vulnerability was detected on. TCP, UDP, ICMP.</td>
</tr>
<tr>
<td>port</td>
<td>(integer) The port number the vulnerability was detected on.</td>
</tr>
<tr>
<td>source</td>
<td>(string) The vulnerability source. HOST, ORACLE, HSSQL, OTHER.</td>
</tr>
</tbody>
</table>
Sample - Search vulnerability instances

Find all vulnerabilities that were previously detected on a host, and that have since been resolved

**API request**

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/hostinstancevuln"
<file.xml
Note: file.xml contains the request POST data
```

**Request POST data**

```xml
<ServiceRequest>
  <filters>
    <Criteria field="hostAssetId" operator="EQUALS">12345</Criteria>
    <Criteria field="found" operator="EQUALS">false</Criteria>
  </filters>
</ServiceRequest>
```

**Response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/hostinstancevuln.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostInstanceVuln>
      <id>9534081</id>
      <hostAssetId>12345</hostAssetId>
      <qid>38167</qid>
      <port>25</port>
      <ssl>true</ssl>
      <found>true</found>
      <ignored>false</ignored>
      <disabled>false</disabled>
      <updated>2012-10-19T21:56:23Z</updated>
      <protocol>TCP</protocol>
      <source>HOST</source>
    </HostInstanceVuln>
  </data>
</ServiceResponse>
```
XSD

<platform_API_server>/qps/xsd/2.0/am/hostinstancevuln.xsd
Count Vulnerabilities

/qps/rest/2.0/count/am/hostinstancevuln

[GET]

Returns the number of host instance vulnerabilities that match the provided criteria. A host instance vulnerability is counted when the asset visible to the user (i.e. it is in the user's scope).

Permissions required - Managers with Full Scope. Users without Full Scope must have these account settings: 1) scope includes the requested asset, and 2) permissions include: Access Permission “API Access” and Asset Management Permission “Read Asset”.

Sample - Count vulnerabilities on assets

Count the number of host instance vulnerabilities across all visible assets

**API request**

```
curl -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/count/am/hostinstancevuln"< file.xml
```

Note: file.xml contains the request POST data

**Request POST data**

```
<ServiceRequest>
  <filters>
    <Criteria field="found" operator="EQUALS">true</Criteria>
  </filters>
</ServiceRequest>
```

**Response**

```
<?xml version="1.0" encoding="UTF-8"?>
```

123
Qualys Asset Management & Tagging API

Host Instance Vulnerability data

XSD

<platform API server>/qps/xsd/2.0/am/hostinstancevuln.xsd
# Host Instance Vulnerability Fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(long) The primary host instance vulnerability key.</td>
</tr>
<tr>
<td>hostAssetId</td>
<td>(long) The ID of the host asset where the vulnerability was found.</td>
</tr>
<tr>
<td>created</td>
<td>(date) The date the vulnerability was added to the KnowledgeBase.</td>
</tr>
<tr>
<td>found</td>
<td>(boolean) Set to true if the QID was detected on the host by the latest scan of that host.</td>
</tr>
<tr>
<td>firstFound</td>
<td>(date) The date/time the vulnerability was first detected on the host.</td>
</tr>
<tr>
<td>lastfound</td>
<td>(date) The most recent date/time the vulnerability was detected on the host.</td>
</tr>
<tr>
<td>lastScanned</td>
<td>(date) The most recent date/time the vulnerability was tested for the host.</td>
</tr>
<tr>
<td>qid</td>
<td>(long) The Qualys vulnerability ID of the vulnerability.</td>
</tr>
<tr>
<td>disabled</td>
<td>(long) Set to true if the QID is marked as disabled in your subscription. Set to false if the QID is not marked disabled.</td>
</tr>
<tr>
<td>fqdn</td>
<td>(string) The fully qualified domain name of the host.</td>
</tr>
<tr>
<td>ssl</td>
<td>(boolean) Set to true if the vulnerability was detected over SSL. Set to false if the vulnerability was not detected over SSL. This element is not returned for information gathered.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>updated</td>
<td>(date) The last date/time the vulnerability data was updated for the host.</td>
</tr>
<tr>
<td>ignored</td>
<td>(boolean) Set to true if the QID/host/port is marked as ignored in your subscription. Set to false if the QID/host/port is not marked ignored.</td>
</tr>
<tr>
<td>protocol</td>
<td>The protocol the vulnerability was detected on (TCP, UDP, ICMP).</td>
</tr>
<tr>
<td>port</td>
<td>(integer) The port number the vulnerability was detected on.</td>
</tr>
<tr>
<td>source</td>
<td>The vulnerability source (HOST, ORACLE, HSSQL, OTHER).</td>
</tr>
</tbody>
</table>
Asset Data Connector

Get Connector Info

/qps/rest/2.0/get/am/assetdataconnector/<id>

[GET]

Returns a single asset data connector by ID.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the asset data connector. Learn more about limiting your results

Permissions required - Managers with Full Scope.

Sample - Fetch asset data connector info

API request

```
curl -n -u "USERNAME:PASSWORD" 
"https://qualysapi.qualys.com/qps/rest/2.0/get/am/assetdataconnector/12345"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/assetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AssetDataConnector>
      <id>12345</id>
      <name>new connector</name>
      <lastSync>2014-11-26T08:44:05Z</lastSync>
      <lastError>Invalid EC2 AuthRecord</lastError>
    </AssetDataConnector>
  </data>
</ServiceResponse>
```
<connectorState>ERROR</connectorState>
<type>AWS</type>
<defaultTags>
    <list>
        <TagSimple>
            <id>1</id>
            <name>EC2</name>
        </TagSimple>
    </list>
</defaultTags>
<activation>
    <ActivationModule>VM</ActivationModule>
</activation>
</AssetDataConnector>
</data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/assetdataconnector.xsd
Update Connector

/qps/rest/2.0/update/am/assetdataconnector

/qps/rest/2.0/update/am/assetdataconnector/<id>

[GET]

Updates writable fields and collections. Only the name and tags can be modified.

Using the NOT EQUALS operator for updating connectors could result in accidental update of unknown connectors without any warning. To prevent accidental updates of unknown connectors, we do not support NOT EQUALS operator for update actions.

Permissions required - Managers with Full Scope.

Sample 1 - Change asset data connector name

Change the name of the asset data connector with ID 12345 and add a tag with the ID of 1 to the defaultTags collection

API request

```
curl -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/update/assetdataconnector/asset/12345" < file.xml
```

Request POST data (file.xml):

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AssetDataConnector>
      <name>Updated Name</name>
      <defaultTags>
        <add>
          <TagSimple>
            <id>1</id>
          </TagSimple>
        </add>
      </defaultTags>
    </AssetDataConnector>
  </data>
</ServiceRequest>
```
Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/assetdataconnector.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <AssetDataConnector>
            <id>12345</id>
            <name>External VPC</name>
            <lastSync>2014-11-26T08:44:05Z</lastSync>
            <lastError/>
            <connectorState>SUCCESS</connectorState>
            <type>AWS</type>
            <defaultTags>
                <list>
                    <TagSimple>
                        <id>1</id>
                        <name>EC2</name>
                    </TagSimple>
                </list>
            </defaultTags>
            <activation>
                <ActivationModule>VM</ActivationModule>
            </activation>
        </AssetDataConnector>
    </data>
</ServiceResponse>

Sample 2 - Add a tag to connectors

Add a tag to all asset data connectors who’s names contain External

API request

Request POST data (file.xml):
    <?xml version="1.0" encoding="UTF-8" ?>
    <ServiceRequest>
       <filters>
         <Criteria field="name" operator="CONTAINS">External</Criteria>
       </filters>
       <data>
         <Asset>
            <tags>
               <add>
                 <TagSimple><id>2</id></TagSimple>
               </add>
            </tags>
         </Asset>
       </data>
    </ServiceRequest>

XML output
    <?xml version="1.0" encoding="UTF-8" ?>
       <responseCode>SUCCESS</responseCode>
       <count>13</count>
       <hasMoreRecords>false</hasMoreRecords>
       <data>
          <AssetDataConnector>
             <id>12345</id>
             <name>External VPC</name>
             <lastSync>2014-11-26T08:44:05Z</lastSync>
             <lastError />
             <connectorState>SUCCESS</connectorState>
             <type>AWS</type>
             <defaultTags>
                <list>
                  <TagSimple>
                     <id>2</id>
                     <name>External</name>
                  </TagSimple>
                </list>
             </defaultTags>
          </AssetDataConnector>
       </data>
    </ServiceResponse>
XSD

<platform_API_server>/qps/xsd/2.0/am/assetdataconnector.xsd
Search Connectors

/qps/rest/2.0/search/am/assetdataconnector

[POST]

Returns a list of asset data connectors that match the provided criteria.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for each asset data connector. Learn more about limiting your results

Pagination - A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.

Permissions required - Managers with Full Scope.

Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(long) Primary key</td>
</tr>
<tr>
<td>name</td>
<td>(string)</td>
</tr>
<tr>
<td>description</td>
<td>(string)</td>
</tr>
<tr>
<td>lastSynch</td>
<td>(date)</td>
</tr>
<tr>
<td>lastError</td>
<td>(string)</td>
</tr>
<tr>
<td>connectorState</td>
<td>(Keyword) PENDING, RUNNING, SUCCESS, ERROR</td>
</tr>
<tr>
<td>activation</td>
<td>(Keyword) AWS</td>
</tr>
</tbody>
</table>
defaultTags.name  (Text) The name of a tag in the defaultTags collection

defaultTag  (Integer) The ID of a tag in the defaultTags collection

disabled  (boolean) Whether execution of the connector is disabled (YES). If disabled the connector will not synchronize assets.

Sample - Find all asset data connectors with tag name USA

API request

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/assetdataconnector" < file.xml
Note: file.xml contains request POST data

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="defaultTags.name" operator="EQUALS">USA</Criteria>
  </filters>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/assetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>13</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AssetDataConnector>
      <id>12345</id>
      <name>DB1</name>
      <lastSync>2014-11-26T08:44:05Z</lastSync>
    </AssetDataConnector>
  </data>
</ServiceResponse>
<lastError />
<connectorState>SUCCESS</connectorState>
<type>AWS</type>
<defaultTags>
  <list>
    <TagSimple>
      <id>3</id>
      <name>USA</name>
    </TagSimple>
  </list>
</defaultTags>
<activation>
  <ActivationModule>VM</ActivationModule>
  <ActivationModule>PC</ActivationModule>
</activation>
</AssetDataConnector>
</data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/assetdataconnector.xsd
Count Connectors

/qps/rest/2.0/count/am/assetdataconnector

[POST]

Returns the number of asset data connectors that match the provided criteria.

Permissions required - Managers with Full Scope.

Sample - Count connectors

Count the number of asset data connectors with the tag name USA

API request

curl -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/count/am/assetdataconnector" < file.xml

Request POST data (file.xml):
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="defaultTags.name" operator="EQUALS">USA</Criteria>
  </filters>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>

XSD
Qualys Asset Management & Tagging API
Asset Data Connector

<platform API server>/qps/xsd/2.0/am/assetdataconnector.xsd
Delete Connector

/qps/rest/2.0/delete/am/assetdataconnector

/qps/rest/2.0/delete/am/assetdataconnector/<id>

[POST]

Delete one or more asset data connectors.

Using the NOT EQUALS operator for deleting connectors could result in accidental deletion of unknown connectors without any warning. To prevent accidental deletion of unknown connectors, we do not support NOT EQUALS operator for delete actions.

Permissions required - Managers with Full Scope.

Sample 1 - Delete a single asset data connector

API request

```bash
curl -n -u "USERNAME:PASSWORD" 
"https://qualysapi.qualys.com/qps/rest/2.0/delete/am/assetdataconnector/12345"
```

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/assetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AssetDataConnector>
      <id>12345</id>
    </AssetDataConnector>
  </data>
</ServiceResponse>
```
Sample 2 - Delete several asset data connectors tagged with the To Delete tag

API request


Request POST data (file.xml):
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">To Delete</Criteria>
  </filters>
</ServiceRequest>

XML output

  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AssetDataConnector>
      <id>1972521</id>
    </AssetDataConnector>
  </data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/assetdataconnector.xsd
Run Connector

/qps/rest/2.0/run/am/assetdataconnector

/qps/rest/2.0/run/am/assetdataconnector/<id>

[POST]

Request that one or more asset data connectors are run. The connectors may be run immediately, or queued to run when there is capacity. The response will almost always indicate that the connector is pending. Use GET calls to monitor the status of connectors.

Permissions required - Managers with Full Scope.

Sample 1 - Run a single connector

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/2.0/run/am/assetdataconnector/12345"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/xsd/2.0/am/assetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AssetDataConnector>
      <id>12345</id>
      <name>DB1</name>
      <lastSync>2014-11-26T08:44:05Z</lastSync>
      <lastError />
      <connectorState>SUCCESS</connectorState>
      <type>AWS</type>
      <defaultTags>
```

140
Sample 2 - Re-run all errored connectors

API request


Request POST data (file.xml):
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="connectorState" operator="EQUALS">ERROR</Criteria>
  </filters>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>13</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AssetDataConnector>
      <TagSimple>
        <id>3</id>
        <name>USA</name>
      </TagSimple>
    </AssetDataConnector>
  </data>
</ServiceResponse>
<id>12345</id>
<name>DB1</name>
<lastSync>2014-11-26T08:44:05Z</lastSync>
...

<AssetDataConnector>
<id>12346</id>
<name>DB2</name>
<lastSync>2015-01-07T01:50:05Z</lastSync>
...

XSD

<platform API server>/qps/xsd/2.0/am/assetdataconnector.xsd
## Connector Fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(long) Primary key, not writeable</td>
</tr>
<tr>
<td>name</td>
<td>(string)</td>
</tr>
<tr>
<td>description</td>
<td>(string)</td>
</tr>
<tr>
<td>lastSynch</td>
<td>(date) Last synch date, not writeable</td>
</tr>
<tr>
<td>lastError</td>
<td>(string) Last error, not writeable</td>
</tr>
<tr>
<td>connectorState</td>
<td>(AssetDataConnectorState) PENDING, RUNNING, SUCCESS, ERROR, not writeable</td>
</tr>
<tr>
<td>type</td>
<td>(AssetDataConnectorType) AWS, not writeable</td>
</tr>
<tr>
<td>defaultTags</td>
<td>(TagSimpleQList) Tags applied to any asset discovered by the connector</td>
</tr>
<tr>
<td>activation</td>
<td>(List&lt;ActivationModule&gt;) Assets discovered by the connector will be activated for the modules specified</td>
</tr>
<tr>
<td>disabled</td>
<td>(boolean) Whether execution of the connector is disabled (YES). If disabled the connector will not synchronize assets.</td>
</tr>
</tbody>
</table>

### Associations

TagSimpleQList - Asset tags to be applied to assets found by the connector. This collection to be added to and removed from by providing a tag ID wrapped in a TagSimple element.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(long) Primary key</td>
</tr>
<tr>
<td>name</td>
<td>(string) Tag name</td>
</tr>
</tbody>
</table>
Get AWS Connector Info

/qps/rest/2.0/get/am/awsassetdataconnector/<id>

[GET]

Returns a single AWS connector by ID.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the AWS connector. Learn more about limiting your results

Permissions required - Managers with full scope.

Sample 1 - Fetch the asset data connector with the ID 12345

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/awsassetdataconnector/12345"

Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AwsAssetDataConnector>
      <id>12345</id>
      <name>new connector</name>
      <lastSync>2014-11-26T09:27:48Z</lastSync>
      <lastError>Invalid EC2 AuthRecord</lastError>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
Sample 2 - Get connector details

Here's how to get details on a connector using GET request. This connector is using ARN. For more information on ARN authentication, refer to Support for Cross-Account Role Authentication.

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/awsassetdataconnector/19201"

Response

<?xml version="1.0" encoding="UTF-8"?>
<count>1</count>
<data>
  <AwsAssetDataConnector>
    <id>19201</id>
    <name>user_john</name>
    <lastSync>2018-02-15T12:51:00Z</lastSync>
    <connectorState>FINISHED_SUCCESS</connectorState>
    <type>AWS</type>
    <defaultTags>
      <list/>
    </defaultTags>
    <activation>
      <list/>
    </activation>
    <disabled>false</disabled>
    <isGovCloudConfigured>false</isGovCloudConfigured>
    <arn>arn:aws:iam::205767712438:role/qualys_dev_test</arn>
    <externalId>1518689351038</externalId>
    <qualysAwsAccountId>383031258652</qualysAwsAccountId>
  </AwsAssetDataConnector>
</data>
</ServiceResponse>

XSD

<platform API server>/qps/xsd/2.0/am/aws_asset_data_connector.xsd
Create AWS Connector

/qps/rest/2.0/create/am/awsassetdataconnector

[POST]

Creates an AWS asset data connector.

disabled (boolean) is used to disable an EC2 connector. This parameter can be set for a “create” or “update” request. When set to “true” the connector is disabled and will not run.

- If a single connector is run and it is disabled an error is returned.

- If multiple connectors are run and all are disabled an error is returned.

- If multiple connectors are run and some are disabled, only connectors that are enabled will run.

Permissions required - Managers with full scope.

Sample 1 - Create new AWS asset data connector

**API request**

```
```

Note: file.xml contains request POST data

**Request POST data**

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>new connector</name>
      <defaultTags>
        <set>
          <TagSimple>
          </TagSimple>
        </set>
      </defaultTags>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>
```
Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>13</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AwsAssetDataConnector>
      <id>12345</id>
      <name>new connector</name>
      <lastSync />
      <lastError />
      <connectorState>PENDING</connectorState>
      <type>AWS</type>
      <defaultTags>
        <list>
          <TagSimple>
            <id>1</id>
            <name>EC2</name>
          </TagSimple>
        </list>
      </defaultTags>
      <activation>
        <ActivationModule>VM</ActivationModule>
      </activation>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
Sample 2 - Create new AWS asset data connector in disabled state

API request

```
```

Request POST data (file.xml):

```
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>conn-disabled</name>
      <activation>
        <set>
          <ActivationModule>VM</ActivationModule>
          <ActivationModule>PC</ActivationModule>
        </set>
      </activation>
      <authRecord>
        <id>90802</id>
      </authRecord>
      <isGovCloudConfigured>false</isGovCloudConfigured>
      <allRegions>true</allRegions>
      <disabled>true</disabled>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>
```
<data>
</ServiceResponse>

XSD

<platform/API server>/qps/xsd/2.0/am/awsassetdataconnector.xsd
Support for AWS GovCloud

/qps/rest/2.0/create/am/awsassetdataconnector

[POST]

Creates an AWS asset data connector for GovCloud.

disabled (boolean) is used to disable an EC2 connector. This parameter can be set for a “create” or “update” request. When set to “true” the connector is disabled and will not run.

- If a single connector is run and it is disabled an error is returned.

- If multiple connectors are run and all are disabled an error is returned.

- If multiple connectors are run and some are disabled, only connectors that are enabled will run.

Permissions required - Managers with full scope.

Sample - Create new AWS asset data connector for GovCloud

API request

Note: file.xml contains request POST data

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
<data>
    <AwsAssetDataConnector>
        <name>gov-cloud</name>
        <activation>
            <set>
                <ActivationModule>VM</ActivationModule>
            </set>
        </activation>
    </AwsAssetDataConnector>
</data>
</ServiceRequest>
< ActivationModule> PC </ActivationModule>
Qualys Asset Management & Tagging API

AWS Asset Data Connector

XSD

<platform_API_server>/qps/xsd/2.0/am/awsassetdataconnector.xsd
Support for China Region

/qps/rest/2.0/create/am/awsassetdataconnector

[POST]

Creates an AWS asset data connector for China region.

You can easily scan EC2 instances included in the AWS China region for vulnerabilities and policy compliance using the Qualys Cloud Platform. You can create/update EC2 connectors to pull instance info from the China region, activate discovered instances for the VM, PC or SCA module, and scan them using our EC2 scan workflow.

isChinaConfigured (boolean) is used to enable the China region for an EC2connector using the AWS Asset Data Connector API (awsassetdataconnector). This parameter can be set for a “create” or “update” request, and is valid only when AWS China option is enabled for your subscription.

When isChinaConfigured is set to “true”

- The connector is configured to pull instance info from the China region only.
- The connector can’t be configured with allRegions set to “true”.

Permissions required - Managers with full scope.

Sample - Create new AWS asset data connector for China region

API request

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/awsassetdataconnector" < file.xml
```

Note: file.xml contains request POST data

Request POST data

```
<?xml version="1.0" encoding="UTF-8" ?>
```
Qualys Asset Management & Tagging API
AWS Asset Data Connector

```xml
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>cn-north-1-conn1</name>
      <authRecord>
        <id>132601</id>
      </authRecord>
      <endpoints>
        <add>
          <AwsEndpointSimple>
            <regionCode>cn-north-1</regionCode>
          </AwsEndpointSimple>
        </add>
      </endpoints>
      <isChinaConfigured>true</isChinaConfigured>
      <disabled>false</disabled>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>136605</id>
      <name>cn-north-1-conn1</name>
      <connectorState>QUEUED</connectorState>
      <type>AWS</type>
      <defaultTags>
        <list/>
      </defaultTags>
      <activation>
        <list/>
      </activation>
      <disabled>false</disabled>
      <isGovCloudConfigured>false</isGovCloudConfigured>
      <isChinaConfigured>true</isChinaConfigured>
      <authRecord>
```
<id>132601</id>
<name>china-auth</name>
</authRecord>
<endpoints>
<list>
  <AwsEndpointSimple>
    <regionCode>cn-north-1</regionCode>
  </AwsEndpointSimple>
</list>
</endpoints>
</AwsAssetDataConnector>
</data>
</ServiceResponse>

XSD

<platform API server>/qps/xsd/2.0/am/awsassetdataconnector.xsd
Support for Cross-Account Role Authentication

/qps/rest/2.0/create/am/awsassetdataconnector

[POST]

Creates an AWS asset data connector using Cross-Account Role Authentication.

Qualys supports the creation of EC2 connectors using a cross-account access role. This allows you to grant Qualys access to your AWS EC2 instances without sharing your AWS security credentials. Qualys will access your AWS EC2 instances by assuming the IAM role that you create in your AWS account.

To get started you’ll need an IAM role created using your AWS account. You can update your existing EC2 connectors to now use cross-account access roles. Note that this migration of your existing EC2 connector to cross account role is unidirectional and cannot be reverted.

You can create only one connector for each unique AWS account. It’s recommended that you merge multiple EC2 connectors into one by removing duplicate connectors before you upgrade to ARN.

Permissions required - Managers with full scope.

Sample 1 - Create a new connector

Create connector when you already have the ARN generated from your AWS account

**API request**

```bash
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/awsassetdataconnector" < file.xml
```

Note: "file.xml" contains the request POST data.

**Request POST data**
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>user_john</name>
      <arn>arn:aws:iam::705355653965:role/ARN_UPGRADE</arn>
      <externalId>1234567890</externalId>
      <endpoints>
        <add>
          <AwsEndpointSimple>
            <regionCode>ap-south-1</regionCode>
          </AwsEndpointSimple>
        </add>
      </endpoints>
      <disabled>false</disabled>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>

Response
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/2.0/am/aws_asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>19803</id>
      <name>user_john</name>
      <connectorState>QUEUED</connectorState>
      <type>AWS</type>
      <defaultTags>
        <list/>
      </defaultTags>
      <activation>
        <list/>
      </activation>
      <disabled>false</disabled>
      <isGovCloudConfigured>false</isGovCloudConfigured>
      <isChinaConfigured>false</isChinaConfigured>
      <arn>arn:aws:iam::705355653965:role/ARN_UPGRADE</arn>
      <externalId>1234567890</externalId>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
Sample 2 - Create a new connector when you want to provide the ARN later

If you have dependencies and cannot provide the ARN at the time of creation, you could always provide the ARN at a later stage. In this case, the AWS connector is created with an INCOMPLETE state.

API request


Note: "file.xml" contains the request POST data.
Request POST data (file.xml):
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>user_john</name>
      <endpoints>
        <add>
          <AwsEndpointSimple>
            <regionCode>ap-south-1</regionCode>
          </AwsEndpointSimple>
        </add>
        <allRegions>false</allRegions>
      </endpoints>
      <disabled>false</disabled>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
XML output

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/2.0/am/aws_asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>19201</id>
      <name>my-aws-connector</name>
      <connectorState>INCOMPLETE</connectorState>
      <type>AWS</type>
      <defaultTags/>
      <activation/>
      <disabled>false</disabled>
      <isGovCloudConfigured>false</isGovCloudConfigured>
      <externalId>1518689351038</externalId>
      <qualysAwsAccountId>383031258652</qualysAwsAccountId>
      <endpoints>
        <list>
          <AwsEndpointSimple>
            <regionCode>ap-south-1</regionCode>
          </AwsEndpointSimple>
        </list>
      </endpoints>
      <allRegions>false</allRegions>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
```

XSD

```xml
<platform_API_server>/qps/xsd/2.0/am/awsassetdataconnector.xsd
```
Update AWS Connector

/qps/rest/2.0/update/am/awsassetdataconnector
/qps/rest/2.0/update/am/awsassetdataconnector/<id>

[POST]

Updates writable fields and collections.

Using the NOT EQUALS operator for updating AWS connectors could result in accidental update of unknown AWS connectors without any warning. To prevent accidental updates of unknown AWS connectors, we do not support NOT EQUALS operator for update actions.

Permissions required - Managers with full scope.

Sample 1 - Update AWS connector name

Change the name of an asset data connector with ID of 12345, add a tag with the ID of 1 to the defaultTags collection, and add us-east-1 as scanned region

**API request**

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/update/awsassetdataconnecto
r/asset/12345" < file.xml
Note: file.xml contains request POST data
```

**Request POST data**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>Updated Name</name>
      <defaultTags>
        <add>
          <TagSimple>
            <id>1</id>
```
### Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/awsassetdataconnector.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <AssetDataConnector>
            <id>12345</id>
            <name>External VPC</name>
            <lastSync>2014-11-26T08:44:05Z</lastSync>
            <lastError/>
            <connectorState>SUCCESS</connectorState>
            <type>AWS</type>
            <defaultTags>
                <list>
                    <TagSimple>
                        <id>1</id>
                        <name>EC2</name>
                    </TagSimple>
                </list>
            </defaultTags>
            <activation>
                <ActivationModule>VM</ActivationModule>
            </activation>
            <disabled>false</disabled>
            <isGovCloudConfigured>false</isGovCloudConfigured>
        </AssetDataConnector>
    </data>
</ServiceResponse>
```
Sample 2 - Update existing key-based connector to cross-account role

Here's how to update an existing connector to use a cross-access account role. You'll need the ARN generated from your AWS account. Note that this migration of your existing EC2 connector to cross account role is unidirectional and cannot be reverted.

API request

curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/update/am/awsassetdataconnector/12345" < file.xml
Note: file.xml contains request POST data

Request POST data

<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <arn>arn:aws:iam::205767712438:role/qualys_dev_test</arn>
      <externalId>123456789</externalId>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/2.0/am/aws_asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>19201</id>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
<platform API server>/qps/xsd/2.0/am/aws_asset_data_connector.xsd
Search AWS Connectors

/qps/rest/2.0/search/am/awsassetdataconnector

[POST]

Returns a list of AWS connectors that match the provided criteria.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for each AWS connector. Learn more about limiting your results

Pagination - A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.

Permissions required - Managers with full scope.

### Searchable fields

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(Integer) Primary key</td>
</tr>
<tr>
<td>name</td>
<td>(Text)</td>
</tr>
<tr>
<td>description</td>
<td>(Text)</td>
</tr>
<tr>
<td>lastSync</td>
<td>(Date)</td>
</tr>
<tr>
<td>lastError</td>
<td>(Text)</td>
</tr>
<tr>
<td>connectorState</td>
<td>(Keyword) PENDING, RUNNING, SUCCESS, ERROR</td>
</tr>
<tr>
<td>activation</td>
<td>(Keyword) VM or PC</td>
</tr>
</tbody>
</table>
Qualys Asset Management & Tagging API

AWS Asset Data Connector

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>defaultTags.name</td>
<td>(Text) The name of a tag in the defaultTags collection</td>
</tr>
<tr>
<td>defaultTag</td>
<td>(Integer) The ID of a tag in the defaultTags collection</td>
</tr>
<tr>
<td>allRegions</td>
<td>(Boolean)</td>
</tr>
<tr>
<td>serviceType</td>
<td>(Keyword) EC2</td>
</tr>
<tr>
<td>endpoint.region</td>
<td>(Text) AWS region code</td>
</tr>
<tr>
<td>authRecord</td>
<td>(Integer) The ID of the authentication record</td>
</tr>
<tr>
<td>authRecord.name</td>
<td>(Text) The name of the authentication record</td>
</tr>
<tr>
<td>disabled</td>
<td>(Boolean) Whether execution of the connector is disabled (YES). If disabled the connector will not synchronize assets.</td>
</tr>
</tbody>
</table>

Sample - Find all asset data connectors with tag name USA

### API request

Note: file.xml contains request POST data

### Request POST data

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="defaultTags.name" operator="EQUALS">USA</Criteria>
  </filters>
</ServiceRequest>
```

### Response
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/xsd/2.0/am/awsassetdataconnector.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>13</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <AwsAssetDataConnector>
            <id>12345</id>
            <name>NEW Connector</name>
            <lastSync>2014-11-26T09:27:48Z</lastSync>
            <lastError>Invalid EC2 AuthRecord</lastError>
            <connectorState>ERROR</connectorState>
            <type>AWS</type>
            <defaultTags>
                <list>
                    <TagSimple>
                        <id>1</id>
                        <name>USA</name>
                    </TagSimple>
                </list>
            </defaultTags>
            <activation/>
            <authRecord>
                <id>1</id>
                <name>my ec2</name>
            </authRecord>
            <endpoints>
                <list>
                    <AwsEndpointSimple>
                        <regionCode>us-east-1</regionCode>
                    </AwsEndpointSimple>
                </list>
            </endpoints>
            <allRegions>false</allRegions>
        </AwsAssetDataConnector>
    </data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/awsassetdataconnector.xsd
Count AWS Connectors

/qps/rest/2.0/count/am/awsassetdataconnector

[POST]

Returns the number of AWS connectors that match the provided criteria.

Permissions required - Managers with full scope.

Sample - Get count of AWS connectors

Count the number of AWS connectors with the tag name USA

**API request**

```bash
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/2.0/count/am/awsassetdataconnector/
"< file.xml
Note: file.xml contains request POST data
```

**Request POST data**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="defaultTags.name" operator="EQUALS">USA</Criteria>
  </filters>
</ServiceRequest>
```

**Response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/awsassetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```
XSD

<platform API server>/qps/xsd/2.0/am/awsassetdataconnector.xsd
Delete AWS Connector

/qps/rest/2.0/delete/am/awsassetdataconnector
/qps/rest/2.0/delete/am/awsassetdataconnector/<id>

[POST]

Delete one or more AWS connectors.

Using the NOT EQUALS operator for deleting AWS connectors could result in accidental deletion of AWS connectors without any warning. To prevent accidental deletion of unknown AWS connectors, we do not support NOT EQUALS operator for delete actions.

Permissions required - Managers with full scope.

Sample 1 - Delete a single AWS connector

API request

```
curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/delete/am/awsassetdataconnector/12345"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/awsassetdataconnector.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <AssetDataConnector><id>12345</id></AssetDataConnector>
    </data>
</ServiceResponse>
```

Sample 2 - Delete several AWS connectors tagged with the To Delete tag
API request

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/delete/am/awsassetdataconne-
tor" < file.xml
Note: file.xml contains request POST data

Request POST data

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="tagName" operator="EQUALS">To Delete</Criteria>
  </filters>
</ServiceRequest>
```

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AssetDataConnector>
      <id>1972521</id>
    </AssetDataConnector>
  </data>
</ServiceResponse>
```

XSD

<platform_API_server>/qps/xsd/2.0/am/awsassetdataconnector.xsd
Run Connector

/qps/rest/2.0/run/am/assetdataconnector

/qps/rest/2.0/run/am/assetdataconnector/<id>

[POST]

Request that one or more asset data connectors are run. The connectors may be run immediately, or queued to run when there is capacity. The response will almost always indicate that the connector is pending. Use GET calls to monitor the status of connectors.

Permissions required - Managers with full scope.

See Run Connector
# AWS Connector Fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(long) Primary key, not writeable</td>
</tr>
<tr>
<td>name</td>
<td>(string)</td>
</tr>
<tr>
<td>description</td>
<td>(string)</td>
</tr>
<tr>
<td>lastSynch</td>
<td>(date) Last synch date, not writeable</td>
</tr>
<tr>
<td>lastError</td>
<td>(string) Last error, not writeable</td>
</tr>
<tr>
<td>connectorState</td>
<td>(AssetDataConnectorState) PENDING, RUNNING, SUCCESS, ERROR, DISABLED, INCOMPLETE, not writeable</td>
</tr>
<tr>
<td>type</td>
<td>(AssetDataConnectorType) AWS, not writeable</td>
</tr>
<tr>
<td>defaultTags</td>
<td>(TagSimpleQList) Tags applied to any asset discovered by the connector</td>
</tr>
<tr>
<td>activation</td>
<td>(List&lt;ActivationModule&gt;) Assets discovered by the connector will be activated for the modules specified</td>
</tr>
<tr>
<td>authRecord</td>
<td>(AwsAuthRecordSimple) The AWS authentication record the connector will use to connect to AWS. When writing/updating it is looked up by the ID field.</td>
</tr>
<tr>
<td>serviceType</td>
<td>(AwsServiceType) EC2</td>
</tr>
<tr>
<td>allRegions</td>
<td>(boolean) If true the end point’s collection will be ignored an all AWS regions scanned</td>
</tr>
<tr>
<td>disabled</td>
<td>(boolean) Whether execution of the connector is</td>
</tr>
</tbody>
</table>
disabled (YES). If disabled the connector will not synchronize assets.

<table>
<thead>
<tr>
<th>arn</th>
<th>Generated by AWS. Ensure that you provide the same ARN that is generated by AWS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>externalId</td>
<td>Random string which is unique for each user.</td>
</tr>
</tbody>
</table>

**Associations**

**AwsEndpointSimpleQList** - A basic wrapper with one field: regionCode. This is the AWS region code, e.g. us-east-1.

**AwsAuthRecordSimple** - The authentication record a connector will use to communicate with AWS. id (long) is the connector ID, and name (string) is a human readable name to identify the connector key.
AWS Authentication Record

Get AWS Auth Record Info

/qps/rest/2.0/get/am/awsauthrecord/<id>

[GET]

Returns a single AWS authentication record by ID.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for the authentication record. Learn more about limiting your results

Permissions required - Managers with full scope.

Sample - Fetch details on AWS authentication record

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/get/am/awsauthrecord/12345"

Response

<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAuthRecord>
      <id>12345</id>
      <name>Auth Record</name>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
    </AwsAuthRecord>
  </data>
</ServiceResponse>
Qualys Asset Management & Tagging API
AWS Authentication Record

</ServiceResponse>

XSD

<platform API server>/qps/xsd/2.0/am/awsauthrecord.xsd
Create AWS Auth Record

/qps/rest/2.0/create/am/awsauthrecord

[POST]

Creates a new authentication record.
Permissions required - Managers with full scope.

Sample - Create new AWS authentication record

API request

curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/awsauthrecord" <
file.xml
Note: file.xml contains request POST data

Request POST data

<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAuthRecord>
      <name>Simple Auth Record</name>
      <description>Production Auth Record</description>
      <accessKeyId>AAAAAAAAAAAAAAAAAA1A</accessKeyId>
      <secretKey>1aA1aa1aaaa1aAaAaaAaa1Aaaa11aaAAAAaaaA</secretKey>
    </AwsAuthRecord>
  </data>
</ServiceRequest>

Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/awsauthrecord.xsd">
<responseCode>SUCCESS</responseCode>
<count>1</count>
<data>
  <AwsAuthRecord>
    <id>12345</id>
    <name>Simple Auth Record</name>
    <description>Production Auth Record</description>
    <created>2014-02-06T19:14:50Z</created>
    <modified>2014-02-06T19:14:50Z</modified>
  </AwsAuthRecord>
</data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/awsauthrecord.xsd
Update AWS Auth Record

/qps/rest/2.0/update/am/awsauthrecord
/qps/rest/2.0/update/am/awsauthrecord/<id>

[POST]

Updates writable fields.

Using the NOT EQUALS operator for updating AWS authentication records could result in accidental update of unknown AWS authentication records without any warning. To prevent accidental updates of unknown AWS authentication records, we do not support NOT EQUALS operator for update actions.

Permissions required - Managers with full scope.

Sample - Update the secret key of AWS auth record

API request

```
Note: file.xml contains request POST data
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAuthRecord>
      <secretKey>1aA1aa1aaaaa1aAaAaaAaa1Aaaaa11aaAAAAaaaA</secretKey>
    </AwsAuthRecord>
  </data>
</ServiceRequest>
```
Response

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/awsauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAuthRecord>
      <id>12345</id>
      <name>Simple Auth Record</name>
      <description>Production Auth Record</description>
      <created>2014-02-06T19:14:50Z</created>
      <modified>2014-02-06T19:14:50Z</modified>
    </AwsAuthRecord>
  </data>
</ServiceResponse>

XSD

<platform_API_server>/qps/xsd/2.0/am/awsauthrecord.xsd
Search AWS Auth Records

/qps/rest/2.0/search/am/awsauthrecord

[POST]

Returns a list of authentication records that match the provided criteria.

Limit your results - Use the optional “fields” parameter to limit the amount of information returned for each authentication record. [Learn more about limiting your results]

Pagination - A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.

Permissions required - Managers with full scope.

Searchable fields

[Click here for available operators]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(Integer) AWS auth record ID</td>
</tr>
<tr>
<td>name</td>
<td>(Text) AWS auth record name</td>
</tr>
<tr>
<td>description</td>
<td>(Text) AWS auth record description</td>
</tr>
<tr>
<td>created</td>
<td>(Date) When AWS auth record was created</td>
</tr>
<tr>
<td>modified</td>
<td>(Date) When AWS auth record was last modified</td>
</tr>
</tbody>
</table>

Sample - Search AWS auth records

Find all authentication records that have a name that contains the string AUTH
Qualys Asset Management & Tagging API
AWS Authentication Record

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/awsaauthrecord" < file.xml
Note: file.xml contains request POST data
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">Simple</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/am/awsaauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>4</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AwsAuthRecord>
      <id>66013771</id>
      <name>Simple Auth Record 1</name>
      <modified>2014-12-22T18:36:44Z</modified>
    </AwsAuthRecord>
    <AwsAuthRecord>
      <id>66023771</id>
      <name>Simple Auth Record 2</name>
      <modified>2014-12-22T18:36:58Z</modified>
    </AwsAuthRecord>
    <AwsAuthRecord>
      <id>66033771</id>
      <name>Simple Auth Record 3</name>
      <modified>2014-12-22T18:37:01Z</modified>
    </AwsAuthRecord>
    <AwsAuthRecord>
      <id>66043771</id>
    </AwsAuthRecord>
  </data>
</ServiceResponse>
```
Qualys Asset Management & Tagging API
AWS Authentication Record

XSD

<platform_API_server>/qps/xsd/2.0/am/awsauthrecord.xsd
Count AWS Auth Records

/qps/rest/2.0/count/am/awsauthrecord

[POST]

Returns the number of authentication records that match the provided criteria.

Permissions required - Managers with full scope.

Sample - Count AWS auth records

Count the number of AWS authentication records that have a name that contains the string AUTH

API request

```bash
curl -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/count/am/awsauthrecord"< file.xml
Note: file.xml contains request POST data
```

Request POST data

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name operator="CONTAINS">AUTH</Criteria>
  </filters>
</ServiceRequest>
```

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```
XSD

<platform_API_server>/qps/xsd/2.0/am/awsauthrecord.xsd
Delete AWS Auth Record

/qps/rest/2.0/delete/am/awsauthrecord
/qps/rest/2.0/delete/am/awsauthrecord/<id>

[POST]

Delete one or more authentication records.

Using the NOT EQUALS operator for deleting AWS authentication records could result in accidental deletion of unknown AWS authentication records without any warning. To prevent accidental deletion of unknown AWS authentication records, we do not support NOT EQUALS operator for delete actions.

Permissions required - Managers with full scope.

Sample 1 - Delete a single authentication record

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/2.0/delete/am/awsauthrecord/12345"

Response

<?xml version="1.0" encoding="UTF-8"?>
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <AssetDataConnector>
            <id>12345</id>
        </AssetDataConnector>
    </data>
</ServiceResponse>
Sample 2 - Delete several authentication records whose names contain the string "delete me"

### API request

```
```

Note: file.xml contains request POST data

### Request POST data

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">delete me</Criteria>
  </filters>
</ServiceRequest>
```

### Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAuthRecord>
      <id>2020094</id>
    </AwsAuthRecord>
  </data>
</ServiceResponse>
```

### XSD

```
<platform_API_server>/qps/xsd/2.0/am/awsauthrecord.xsd
```
## AWS Auth Record Fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(string) Name of the authentication record</td>
</tr>
<tr>
<td>description</td>
<td>(string) Brief description of the authentication record</td>
</tr>
<tr>
<td>created</td>
<td>(date) When record was created, not writeable</td>
</tr>
<tr>
<td>modified</td>
<td>(date) When records was last modified, not writeable</td>
</tr>
<tr>
<td>secretKey</td>
<td>(string) The AWS secret key - write only, cannot be read</td>
</tr>
<tr>
<td>accessKeyId</td>
<td>(string) The AWS access key - write only, cannot be read</td>
</tr>
</tbody>
</table>