



# Asset Management and Tagging API v2

User Guide  
Version 2.33

June 18, 2018

Copyright 2013-2018 by Qualys, Inc. All Rights Reserved.

Qualys, the Qualys logo and QualysGuard are registered trademarks of Qualys, Inc. All other trademarks are the property of their respective owners.

Qualys, Inc.  
919 E Hillsdale Blvd  
4th Floor  
Foster City, CA 94404  
1 (650) 801 6100

# CONTENTS

---

## **Preface**

### **Chapter 1 Welcome**

Get Started .....	6
Get API Notifications .....	6
Introduction to the API Framework .....	7
Request URL .....	7
URL to the Qualys API Server .....	7
Authentication .....	9
JSON Support .....	9
Usage .....	10
Know your Portal Version .....	13

### **Chapter 2 Tag API**

Get Tag Info .....	16
Create Tag .....	18
Get to know Groovy .....	20
Update Tag .....	21
Search Tags .....	24
Count Tags .....	27
Delete Tag .....	29
Evaluate Tag .....	31
List Users with their Tags .....	34
Tag Fields .....	38

### **Chapter 3 Host Asset API**

Get Host Asset Info .....	40
Create Host Asset .....	47
Update Host Asset .....	54
Search Hosts Assets .....	59
Count Host Assets .....	65
Delete Host Asset .....	67
Activate Host Asset .....	69
Host Asset Fields .....	72

### **Chapter 4 Asset API**

Get Asset Info .....	75
Update Asset .....	77
Search Assets .....	81
Count Assets .....	84
Delete Asset .....	86

Activate Asset ..... 88  
Asset Fields ..... 91

**Chapter 5 Host Instance Vulnerability API**

Get Vulnerability Info..... 93  
Search Vulnerabilities..... 95  
Count Vulnerabilities ..... 98  
Host Instance Vulnerability Fields ..... 99

**Chapter 6 Asset Data Connector API**

Get Connector Info..... 101  
Update Connector ..... 103  
Search Connectors..... 107  
Count Connectors ..... 110  
Delete Connector..... 111  
Run Connector..... 114  
Connector Fields ..... 117

**Chapter 7 AWS Asset Data Connector API**

Get AWS Connector Info ..... 119  
Create AWS Connector ..... 122  
Support for AWS GovCloud ..... 127  
Support for China Region..... 129  
Support for Cross-Account Role Authentication ..... 132  
Update AWS Connector..... 136  
Search AWS Connectors ..... 140  
Count AWS Connectors ..... 143  
Delete AWS Connector..... 144  
Run Connector..... 146  
AWS Connector Fields ..... 147

**Chapter 8 AWS Authentication Record API**

Get AWS Auth Record Info ..... 150  
Create AWS Auth Record ..... 151  
Update AWS Auth Record ..... 153  
Search AWS Auth Records ..... 155  
Count AWS Auth Records..... 158  
Delete AWS Auth Record ..... 159  
AWS Auth Record Fields ..... 162

**Appendix A JSON Support**

---

# Preface

Using the Qualys Asset Management and Tagging API, third parties can integrate the Qualys Security and Compliance solution into their own applications using an extensible XML interface. This user guide is intended for application developers who will use the Qualys Asset Management and Tagging API.

## About Qualys

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

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

## Contact Qualys Support

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

# CHAPTER 1

---

## Welcome

Welcome to Qualys Asset Management and Tagging API. The sections that follow describe how to use the Tags API, Host Asset API, Asset API and the Host Instance Vulnerability API.

### Get Started

[Introduction to the API Framework](#) - We recommend you review important information about the API framework.

[URL to the Qualys API Server](#) - We'll give you the basics about making API requests. The base URL depends on the platform where your Qualys account is located.

[Authentication](#) - We'll tell you about the method used for authentication. API requests must authenticate using Qualys credentials.

[Usage](#) - Check out the basics about making requests and receiving data from our asset management application.

[Know your Portal Version](#)

### Get API Notifications

We recommend you join our Community and subscribe to our API notifications so you'll get email notifications telling you about important upcoming API enhancements and changes.

#### From our Community

[Join our Community](#)

[Subscribe to API Notifications \(select Receive email notifications\)](#)

# Introduction to the API Framework

The Qualys Asset Management and Tagging API framework offers numerous innovations and new functionality compared to the other Qualys API frameworks.

## Request URL

The URL for making Asset Management and Tagging API requests respects the following structure:

`https://<baseurl>/qps/rest/2.0/<operation>/<module>/<object>/<object_id>`

where the components are described below.

<code>&lt;baseurl&gt;</code>	The Qualys API server URL that you should use for API requests depends on the platform where your account is located. The base URL for Qualys US Platform 1 is: <code>https://qualysapi.qualys.com</code>
<code>&lt;operation&gt;</code>	The request operation, such as get a list, get a count, search, create, and update.
<code>&lt;module&gt;</code>	The API module. For the Asset Management and Tagging API, the module is: "am".
<code>&lt;object&gt;</code>	The module specific object.
<code>&lt;object_id&gt;</code>	(Optional) The module specific object ID, if appropriate.

## URL to the Qualys API Server

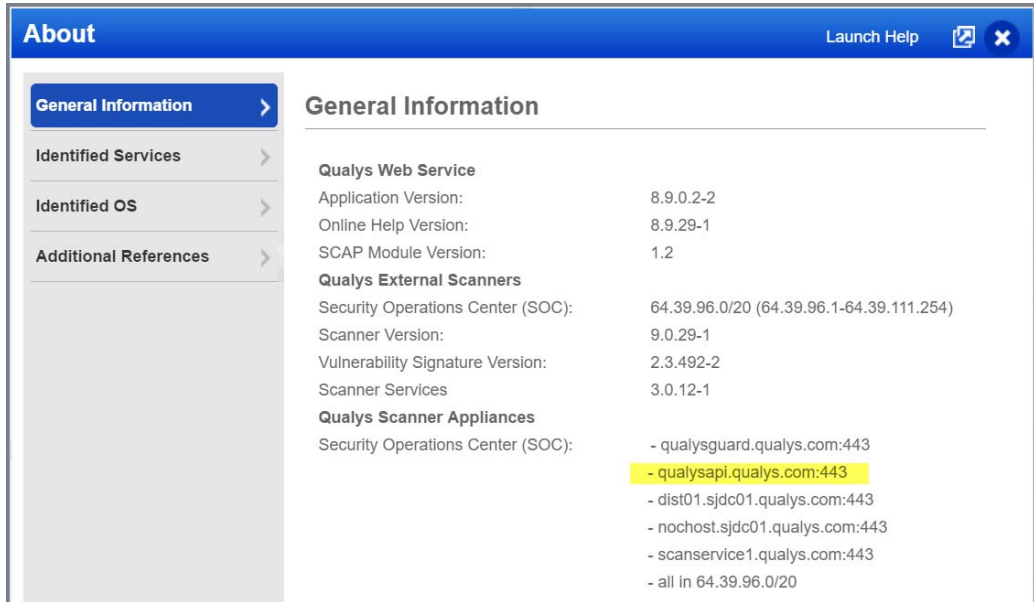
Qualys maintains multiple Qualys platforms. The Qualys API server URL that you should use for API requests depends on the platform where your account is located.

<b>Account Location</b>	<b>API Server URL</b>
Qualys US Platform 1	<code>https://qualysapi.qualys.com</code>
Qualys US Platform 2	<code>https://qualysapi.qg2.apps.qualys.com</code>
Qualys US Platform 3	<code>https://qualysapi.qg3.apps.qualys.com</code>
Qualys EU Platform 1	<code>https://qualysapi.qualys.eu</code>
Qualys EU Platform 2	<code>https://qualysapi.qg2.apps.qualys.eu</code>
Qualys India Platform 1	<code>https://qualysapi.qg1.apps.qualys.in</code>
Qualys Private Cloud Platform	<code>https://qualysapi.&lt;customer_base_url&gt;</code>

The Qualys API documentation and sample code use the API server URL for the Qualys US Platform 1. If your account is located on another platform, please replace this URL with the appropriate server URL for your account.

**Still have questions?** You can easily find the API server URL for your account.

Just log in to your Qualys account and go to Help > About. You'll see this information under Security Operations Center (SOC).



The screenshot shows the 'About' page in a web browser. The page has a blue header with the title 'About' and a 'Launch Help' button. On the left, there is a sidebar with navigation options: 'General Information' (selected), 'Identified Services', 'Identified OS', and 'Additional References'. The main content area is titled 'General Information' and contains the following details:

Qualys Web Service	
Application Version:	8.9.0.2-2
Online Help Version:	8.9.29-1
SCAP Module Version:	1.2

Qualys External Scanners	
Security Operations Center (SOC):	64.39.96.0/20 (64.39.96.1-64.39.111.254)
Scanner Version:	9.0.29-1
Vulnerability Signature Version:	2.3.492-2
Scanner Services	3.0.12-1

Qualys Scanner Appliances	
Security Operations Center (SOC):	- qualysguard.qualys.com:443
	- qualysapi.qualys.com:443
	- dist01.sjdc01.qualys.com:443
	- nochostr.sjdc01.qualys.com:443
	- scanservice1.qualys.com:443
	- all in 64.39.96.0/20



## Authentication

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

For more 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.

### Example

Basic authentication - recommended option:

```
curl -u "USERNAME:PASSWORD"  
https://qualysapi.qualys.com/qps/rest/2.0/count/am/hostasset
```

where portal.qualys.com is the base URL to the Qualys API server where your account is located.

## JSON Support

Qualys Asset Management and Tagging API supports JSON requests and responses.

[Learn more](#)

# Usage

## 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.

Element	Purpose
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 XML tags

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

Invalid tag	Valid tag
This syntax will not work: <id> 34234 </id>	This syntax will work just fine: <id>345254</id>

## 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

Name	Description
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.

### Example

```
<?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.

### Example - Get Request

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

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

id,name,created

### **Example - Search Request**

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

```
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:

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

# Know your Portal Version

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

**URL:** `https://qualysapi.qualys.com/qps/rest/portal/version`

**Methods allowed:** GET

## Examples

### Example 1: XML

#### API Request:

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Accept: application/xml"  
https://qualysapi.qualys.com/qps/rest/portal/version
```

#### 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/versi  
on.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Portal-Version>  
      <PortalApplication-VERSION>2.33.0.0-SNAPSHOT-1 DEVELOP #352  
(2018-05-07T22:53:43Z)</PortalApplication-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>
```

## Example 2: JSON

### API Request:

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Accept: application/json"  
https://qualysapi.qualys.com/qps/rest/portal/version
```

### Response:

```
{  
  "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  
  }  
}
```

## Tag API

The Tag API provides a suite of API functions for creating and managing tags. Also the related Admin User API lets you list users and their associated tags.

### **Operations**

[Get Tag Info](#)

[Create Tag](#)

[Update Tag](#)

[Search Tags](#)

[Count Tags](#)

[Delete Tag](#)

[Evaluate Tag](#)

[List Users with their Tags](#)

### **Reference**

[Tag Fields](#)

## Get Tag Info

Returns a single tag by ID.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/get/am/tag/<id>
<b>Method</b>	GET
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for the tag. <a href="#">Learn more</a>

## Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:  
- Access Permission “API Access”

## Example

Fetch tag ID 12345.

### 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/xs  
d/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/>  
</children>  
</Tag>  
</data>  
</ServiceResponse>
```

## Create Tag

Create a new tag and possibly child tags.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/create/am/tag
<b>Method</b>	POST

## Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Tag Permission "Create User Tag"
- Tag Permission "Modify Dynamic Tag Rules" (to create a dynamic tag)

## Example

This example creates a new tag and 3 child tags.

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 (file.xml):

```
<?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/xs
d/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>
<TagSimple>
  <id>3</id>
  <name>Child 3</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.

### From our Community

[Create Asset Tags using Groovy](#)

# Update Tag

Update fields for a tag and collections of tags.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/update/am/tag/<id> or https://<baseurl>/qps/rest/2.0/update/am/tag
<b>Method</b>	POST

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

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Tag Permission "Edit User Tag"
- Tag Permission "Modify Dynamic Tag Rules" (to update a dynamic tag)

## Example

### Request:

This example renames parent tag 12345 and removes some of its child tags.

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

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

### Request POST data (file.xml):

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

```
        <TagSimple><id>123</id></TagSimple>
        <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/xs
d/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>
    <TagSimple>
        <id>456</id>
        <name>Linked Child 2</name>
    </TagSimple>
</list>
</children>
</Tag>
</data>
</ServiceResponse>
```

## Search Tags

Returns a list of tags that match the provided criteria.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/search/am/tag
<b>Method</b>	POST
<b>Pagination</b>	A maximum of 100 tags are returned by default. To customize this specify a “preferences” tag in the POST body of your request.
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for each tag. <a href="#">Learn more</a>

## Searchable Fields

These fields can be used to search for tags.

id (Integer)	ruleType(Keyword:STATIC,GROOVY, OS_REGEX, NETWORK_RANGE, NAME_CONTAINS, INSTALLED_SOFTWARE, OPEN_PORTS, VULN_EXIST, ASSET_SEARCH, CLOUD_ASSET)
name (Text)	color (Text formatted as #FFFFFF where F can be any value between 0-9 and A-F)
parentTagId (Integer)	

### Allowed Operators

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



## Permissions

Managers with Full Scope

Users without Full Scope must have this account permission:

- Access Permission “API Access”

## Example

Find all tags with Groovy Script tag rules.

### Request:

```
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
```

### Request POST data (file.xml):

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

# Count Tags

Count all the children of a tag.

**URL**                                https://<baseurl>/qps/rest/2.0/count/am/tag  
**Method**                             POST

## Fields

These fields can be used to count tags.

id (Integer)	ruleType(Keyword:STATIC,GROOVY, OS_REGEX, NETWORK_RANGE, NAME_CONTAINS, INSTALLED_SOFTWARE, OPEN_PORTS, VULN_EXIST, ASSET_SEARCH, CLOUD_ASSET)
name (Text)	color (Text formatted as #FFFFFF where F can be any value between 0-9 and A-F)
parentTagId (Integer)	

### Allowed Operators

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

## Permissions

Managers with Full Scope

Users without Full Scope must have this account permission:

- Access Permission "API Access"

## Example

Get a count of all the children of tag ID 12345

Request:

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

Request POST data (file.xml):

```
<ServiceRequest>
  <filters>
    <Criteria field="parent" operator="EQUALS">12345</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/xs
d/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
</ServiceResponse>
```

# Delete Tag

Delete tags one or more tags.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/delete/am/tag/<id> or https://<baseurl>/qps/rest/2.0/delete/am/tag
<b>Method</b>	POST

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

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Tag Permission "Delete User Tag"

## Examples

Delete the tag that has the ID 12345.

### Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/delete/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/xs
d/2.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <SimpleTag>
      <id>12345</id>
    </SimpleTag>
```

```
</data>  
</ServiceResponse>
```

# Evaluate Tag

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.

**URL**                    `https://<baseurl>/qps/rest/2.0/evaluate/am/tag/<id>`  
or  
`https://<baseurl>/qps/rest/2.0/evaluate/am/tag`

**Method**                POST

## Fields

These fields can be used to evaluate tags.

id (Integer)	ruleType(Keyword:STATIC,GROOVY, OS_REGEX, NETWORK_RANGE, NAME_CONTAINS, INSTALLED_SOFTWARE, OPEN_PORTS, VULN_EXIST, ASSET_SEARCH, CLOUD_ASSET)
name (Text)	color (Text formatted as #FFFFFF where F can be any value between 0-9 and A-F)
parentTagId (Integer)	

### Allowed Operators

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

## Permissions

Managers with Full Scope

Users without Full Scope must have this account permission:

- Access Permission "API Access"

## Examples

Evaluate all tags that have Groovy Script tag rules.

### 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"
```

### Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>  
<ServiceRequest>  
  <filters>  
    <Criteria field="ruleType"  
operator="EQUALS">GROOVY</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/xs  
d/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>
  <list>
    <SimpleTag>
      <id>123</id>
      <name>Red</name>
    </SimpleTag>
  </list>
</children>
</Tag>
</data>
</ServiceResponse>
```

## List Users with their Tags

The Admin User API (/qps/rest/1.0/{action}/admin/user) gives 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.

**URL:** `https://qualysapi.qualys.com/qps/rest/1.0/{action}/admin/user`

**Method allowed:** POST, GET

### Search Users (POST)

You can 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.

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 (file.xml):

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

#### XML 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/1.0/a
dmin/user.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>>false</hasMoreRecords>
  <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>QWEB_PCI</Module>
    <Module>WAS</Module>
    <Module>ADMIN</Module>
    <Module>ASSET_MANAGEMENT</Module>
    <Module>QWEB_VM</Module>
  </list>
</modules>
</User>
</data>
</ServiceResponse>
```

### **Count Users (POST)**

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

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 (file.xml):

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

#### XML 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/1.0/admin/user.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
</ServiceResponse>
```

### Get details of a User

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

XSD: User.XSD

#### API Request:

```
curl -u "USERNAME:PASSWORD" " -X GET -H "Content-type: text/xml"  
"https://qualysapi.qualys.com/qps/rest/1.0/get/admin/user/3989626" <
```

#### XML 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/1.0/admin/user.xsd">  
  <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>
```

## Tag Fields

Writable	Read only
parentTagId (Integer)	created (Date)
color (Text)	modified (Date)
ruleText (Text)	
srcAssetGroupId (Integer)	
srcBusinessUnitId (Integer)	
srcOperatingSystemName (Text)	
children (TagSimpleQList)	

## 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	
id	long (tag primary key)
name	string (tag name)

## Host Asset API

The Host Asset API provides a suite of API functions for managing host assets. In many cases these are hosts detected by our cloud scanners. Host assets can also be added manually by the Qualys API or user interface. The HostAsset members identify operating system, NetBIOS, tags, open ports, NICs, installed software, EC2 source information and current vulnerabilities (all instances).

### Operations

[Get Host Asset Info](#)

[Create Host Asset](#)

[Update Host Asset](#)

[Search Hosts Assets](#)

[Count Host Assets](#)

[Delete Host Asset](#)

[Activate Host Asset](#)

### Reference

[Host Asset Fields](#)

## Get Host Asset Info

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

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/get/am/hostasset/&lt;id&gt;</code>
<b>Method</b>	GET
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for the host asset. <a href="#">Learn more</a>

## Permissions

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”
- Asset Management Permission “Read Asset”

## Example

### Example 1: Fetch host asset ID and list asset details

#### 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/xs  
d/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>2016-02-12T06:21:54Z</created>
```



```

<modified>2016-02-13T01:14:34Z</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>
          <version>1</version>
        </HostAssetSoftware>
        <HostAssetSoftware>
          <name>Security Update for Windows XP
(KB950974)</name>
          <version>1</version>
        </HostAssetSoftware>
      </list>
    </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>
    <HostAssetVolume>
        <name>C:</name>
        <size>16106090496</size>
        <free>2418925568</free>
    </HostAssetVolume>
</list>
</volume>
<account>
    <list>
        <HostAssetAccount>
            <username>Administrator</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>Guest</username>
        </HostAssetAccount>
    </list>
</account>
<networkInterface>
    <list>
        <HostAssetInterface>
            <hostname>XPSP2-32-27-145</hostname>
            <interfaceName>VMware Accelerated AMD PCNet
Adapter - Packet Scheduler Miniport</interfaceName>
            <macAddress>00:50:56:A9:46:72</macAddress>
            <type>LOCAL</type>
            <address>10.10.23.245</address>
            <gatewayAddress>10.10.23.1</gatewayAddress>
        </HostAssetInterface>
    </list>
</networkInterface>
</HostAsset>
</data>
</ServiceResponse>

```

### Example 2: Fetch host asset ID of AWS EC2 asset and list asset details

#### Request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostasset/709838"
```

#### Response:

Tags for the EC2 asset appear in the **<Ec2AssetSourceSimple>** element, shown in bold below.

```
<?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/a  
m/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>  
                </list>  
              </tags>  
            </ec2InstanceTags>  
          </list>  
        </sourceInfo>  
      </HostAsset>  
    </data>  
  </ServiceResponse>  
</xml>
```

```

        <EC2Tags>
          <key>JIRA</key>
          <value>POR-6719</value>
        </EC2Tags>
        <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-7bbbcd56</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>
<vuln>
  <list/>
</vuln>
</processor>

```

### Chapter 3 — Host Asset API

#### Get Host Asset Info

```
        <list/>
    </processor>
    <volume>
        <list/>
    </volume>
    <account>
        <list/>
    </account>
    <networkInterface>
        <list>
            <HostAssetInterface>
                <interfaceId>eni-09f901fe</interfaceId>
                <interfaceName>Primary network
interface</interfaceName>
                <type>PRIVATE</type>
                <address>10.97.15.117</address>
            </HostAssetInterface>
        </list>
    </networkInterface>
</HostAsset>
</data>
</ServiceResponse>
```

## Create Host Asset

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? Once you've created new host assets you need to activate the assets in order to make them available for scanning and reporting in the Vulnerability Management (VM) module. See [Activate Host Asset](#)

<b>URL</b>	https://<baseurl>/qps/rest/2.0/create/am/hostasset
<b>Method</b>	POST

## Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Asset Management Permission "Create Asset"

## Example

### Example 1: Create new host asset with tags

#### Request:

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

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

#### Request POST data (file.xml):

```
<?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>
<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/xs
d/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>2020094</id>
      <name>My Windows Asset</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>
<sourceInfo>
    <list/>
</sourceInfo>
<os>Windows 7</os>
<dnsHostName>localhost</dnsHostName>
<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>

```

### Example 2: Bulk creation of assets

#### Request:

```

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

```

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

POST data:

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

```

    </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>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>
<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>
        <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>

```

## Update Host Asset

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

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/update/am/hostasset/&lt;id&gt;</code> > or <code>https://&lt;baseurl&gt;/qps/rest/2.0/update/am/hostasset</code>
<b>Method</b>	POST

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

## Permissions

Managers with Full Scope

Users without Full Scope must have these account settings: 1) scope includes the requested asset(s), and 2) permissions include:

- Access Permission "API Access"
- Asset Management Permission "Update Asset"

## Example

### Request 1:

Update some of the fields for host asset ID 12345.

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

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

### Request POST data (file.xml):

```
<?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>2014-02-06T19:16:35Z</created>
            <modified>2014-02-06T19:16:35Z</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>

```

```
    </vuln>  
  </HostAsset>  
</data>  
</ServiceResponse>
```

Request 2:

Update some of the fields for host assets that have names containing the word OLD.

```
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 (file.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</name>  
        </HostAssetSoftware>  
      </set>  
    </software>  
    <openPort>  
      <add>  
        <HostAssetOpenPort>
```



```

        <port>8080</port>
        <protocol>TCP</protocol>
      </HostAssetOpenPort>
    </add>
  </openPort>
</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>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>
      <version>95</version>
    </HostAssetSoftware>
  </list>
</software>
<vuln>
  <list/>
</vuln>
</HostAsset>
</data>
</ServiceResponse>
```

## Search Hosts Assets

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).

<b>URL</b>	https://<baseurl>/qps/rest/2.0/search/am/hostasset
<b>Method</b>	POST
<b>Pagination</b>	A maximum of 100 host assets are returned by default. To customize this specify a “preferences” tag in the POST body of your request.
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for each host asset. <a href="#">Learn more</a>

### Searchable Fields

These fields can be used to search for host assets.

qwebHostId (Integer)	netbiosName (Text)
lastVulnScan (Date)	netbiosNetworkID (Text)
lastComplianceScan (Date)	networkGuid (Text)
informationGatheredUpdated (Date)	trackingMethod (Keyword: NONE, IP, DNSNAME, NETBIOS, INSTANCE_ID, QAGENT)
os (Text)	port (Integer)
dnsHostName (Text)	installedSoftware (Text)
address(Text)	tagName (Text)
vulnsUpdated(Date)	tagId (Integer)
id (Integer)	netbiosName (Text)
name (Text)	netbiosNetworkId (Text)
created (Date)	update (Date)
type (Text)	agentVersion (String)
activationKey (String)	agentConfigurationId (Long)
agentConfigurationName (String)	lastCheckedIn (Date)

Allowed Operators

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

These fields can be used to search for cloud agents installed on host assets.

<b>Parameter</b>	<b>Allowed Operators</b>
agentVersion (String)	EQUALS, LESSER, GREATER
activationKey (String)	EQUALS
agentConfigurationId (Long)	EQUALS
agentConfigurationName (String)	EQUALS, CONTAINS
lastCheckedIn (Date)	EQUALS, LESSER, GREATER

## Permissions

### Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Asset Management Permission "Read Asset"

## Example

### Request 1:

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

```
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
```

### Request POST data (file.xml):

```
<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"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/2.0/am/hostasset.xsd">
  <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>2014-02-06T19:16:35Z</created>
      <modified>2014-02-06T19:16:35Z</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>INSTANCE_ID</trackingMethod>
<openPort>
  <list/>
</openPort>
<software>
  <list/>
</software>
<vuln>
  <list/>
</vuln>
</HostAsset>
</data>
</ServiceResponse>
```

**Request 2:**

Find cloud agents with a specific agent version.

```
curl -u fo_username:password -X POST -H "X-Requested-With: curl" -
H "Content-Type: text/xml" -H "Cache-Control: no-cache" --data-
binary @host_asset_search.xml
"http://qualysapi.qualys.com/qps/rest/2.0/search/am/hostasset/"
```

**Request POST data (host\_asset\_search.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:**

```
<?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/hostasset.xsd">
  <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-  
            86fad2dda4f4</activationId>  
        <title>harshal</title>  
    </activationKey>  
</agentInfo>  
<netbiosName>102115-M83</netbiosName>  
<address>10.100.11.163</address>  
<trackingMethod>QAGENT</trackingMethod>  
<openPort>  
    <list/>  
</openPort>  
<software>  
    <list/>  
</software>  
<vuln>  
    <list/>  
</vuln>  
<processor>  
    <list/>  
</processor>  
<volume>  
    <list/>  
</volume>  
<account>  
    <list/>  
</account>  
<networkInterface>  
    <list/>  
</networkInterface>  
</HostAsset>  
</data>  
</ServiceResponse>
```



# Count Host Assets

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).

**URL**                                https://<baseurl>/qps/rest/2.0/count/am/hostasset  
**Method**                            POST

## Fields

These fields can be used to count host assets.

qwebHostId (Integer)	netbiosName (Text)
lastVulnScan (Date)	netbiosNetworkID (Text)
lastComplianceScan (Date)	networkGuid (Text)
informationGatheredUpdated (Date)	trackingMethod (Keyword: NONE, IP, DNSNAME, NETBIOS, INSTANCE_ID, QAGENT)
os (Text)	port (Integer)
dnsHostName (Text)	installedSoftware (Text)
address(Text)	tagName (Text)
vulnsUpdated(Date)	tagId (Integer)
id (Integer)	netbiosName (Text)
name (Text)	netbiosNetworkId (Text)
created (Date)	update (Date)
type (Text)	

### Allowed Operators

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

## Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission “API Access”
- Asset Management Permission “Read Asset”

## Example

### Request:

```
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 (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>  
<ServiceRequest>  
  <filters>  
    <Criteria field="os" operator="EQUALS">Windows</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/xs  
d/2.0/am/hostasset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>235</count>  
</ServiceResponse>
```

# Delete Host Asset

Delete one or more host assets.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/delete/am/hostasset/<id> or https://<baseurl>/qps/rest/2.0/delete/am/hostasset
<b>Method</b>	POST

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

Managers with Full Scope

Users without Full Scope must have these account settings: 1) scope includes the requested asset(s), and 2) permissions include:

- Access Permission "API Access"
- Asset Management Permission "Delete Asset"

## Example

Delete host assets with the tag "To Delete".

### 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 (file.xml):

```
<?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/xs
d/2.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>2020094</id>
    </HostAsset>
  </data>
</ServiceResponse>
```

# Activate Host Asset

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) application module and/or the Policy Compliance (PC) module.

**URL for VM assets**    `https://<baseurl>/qps/rest/2.0/activate/am/hostasset/<id>?module=QWEB_VM`  
or  
`https://<baseurl>/qps/rest/2.0/activate/am/hostasset?module=QWEB_VM`

**URL for PC assets**    `https://<baseurl>/qps/rest/2.0/activate/am/hostasset/<id>?module=QWEB_PC`  
or  
`https://<baseurl>/qps/rest/2.0/activate/am/hostasset?module=QWEB_PC`

**Method**                    POST

## Fields

These fields can be used to activate host assets.

qwebHostId (Integer)	netbiosName (Text)
lastVulnScan (Date)	netbiosNetworkID (Text)
lastComplianceScan (Date)	networkGuid (Text)
informationGatheredUpdated (Date)	trackingMethod (Keyword: NONE, IP, DNSNAME, NETBIOS, INSTANCE_ID, QAGENT)
os (Text)	port (Integer)
dnsHostName (Text)	installedSoftware (Text)
address(Text)	tagName (Text)
vulnsUpdated(Date)	tagId (Integer)
id (Integer)	netbiosName (Text)
name (Text)	netbiosNetworkId (Text)
created (Date)	update (Date)
type (Text)	

## Chapter 3 — Host Asset API

### Activate Host Asset

#### Allowed Operators

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

## Permissions

### Managers with Full Scope

Users without Full Scope must have these account settings: 1) scope includes the requested asset(s), and 2) permissions include:

- Access Permission “API Access”

For PC assets, the Policy Compliance (PC) module must be enabled

## Example

Activate the host assets with the tag “Export to VM”.

### Request:

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

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

### Request POST data (file.xml):

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

See also [Activate Asset](#)

## Host Asset Fields

Writable	Read only
os (string)	qwebHostId (long)
dnsHostName (string)	lastVulnScan (date)
netbiosName (string)	lastComplianceScan (date)
netbiosNetworkId (integer)	vulnsUpdated (date)
networkGuid (uuid)	informationGatheredUpdated (date)
address (string)	account (HostAssetAccount)
trackingMethod (AssetTrackingMethod: NONE, IP, DNSNAME, NETBIOS, INSTANCE_ID, QAGENT)	biosDescription (string)
openPort (HostAssetOpenPortQList)	manufacturer (string)
software (HostAssetSoftwareQList)	model (string)
	networkInterface (HostAssetInterface)
	processor (HostAssetProcessor)
	timezone (string)
	totalMemory (long)
	volume (HostAssetVolume)

## Associations

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

### HostAssetOpenPortList

port	integer
protocol	Protocol (TCP, UDP, ICMP)



### HostAssetOpenPortList

---

serviceId	integer
serverName	string (name of the service detected on the port - read only)

---

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

### HostAssetSoftware

---

name (string)
version (string)

---

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.

### HostInstanceVuln

---

qid (long)
hostInstanceVulnID (long)
firstFound (date)
lastFound (date)

---

## Asset API

The Asset API is a subset of the Host Asset API. The Asset members identify name, tags, and EC2 source information.

### **Operations**

[Get Asset Info](#)

[Update Asset](#)

[Search Assets](#)

[Count Assets](#)

[Delete Asset](#)

[Activate Asset](#)

### **Reference**

[Asset Fields](#)

# Get Asset Info

Returns a single asset by ID.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/get/am/asset/<id>
<b>Method</b>	GET
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for the asset. <a href="#">Learn more</a>

## Permissions

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”
- Asset Management Permission “Read Asset”

## Example

This example fetches the asset ID 12345 and lists asset details.

### 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"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/2.0/am/asset.xsd">  
  <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>
```

```
<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>
```

# Update Asset

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

<b>URL</b>	https://<baseurl>/qps/rest/2.0/update/am/asset/<id> or https://<baseurl>/qps/rest/2.0/update/am/asset
<b>Method</b>	POST

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

Managers with Full Scope

Users without Full Scope must have these account settings: 1) scope includes the requested asset(s), and 2) permissions include:

- Access Permission "API Access"
- Asset Management Permission "Update Asset"

## Example

### Request 1:

Update tag 12345 and give it another name.

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

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

### Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <Asset>
      <name>Updated Name</name>
    </Asset>
```

```
</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/xs  
d/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>
```

Request 2:

Update tags that have tag names containing the word DELETED.

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

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

Request POST data (file.xml):

```
<?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>
```

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/xs
d/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>
```



## Search Assets

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).

<b>URL</b>	https://<baseurl>/qps/rest/2.0/search/am/asset
<b>Method</b>	POST
<b>Pagination</b>	A maximum of 100 assets are returned by default. To customize this specify a “preferences” tag in the POST body of your request.
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for each asset. <a href="#">Learn more</a>

## Searchable Fields

These fields can be used to search for assets.

id (Integer)	type (Keyword: UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN)
name (Text)	tagName (Text)
created (Date)	tagId (Text)
updated (Date)	solrQuery (Text, an asset solr query)

### Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

For tagName and tagID, parents of the tag will also match. For example if tag A had child tag B and an asset has tag A, then a search for tag B will return assets with tags A and B.

## Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Asset Management Permission "Read Asset"

## Example

Find an asset with a particular tag.

Request:

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

Request POST data (file.xml):

```
<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/xs
d/2.0/am/asset.xsd">
  <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>
```

## Count Assets

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).

**URL** `https://<baseurl>/qps/rest/2.0/count/am/asset`  
**Method** `POST`

### Fields

These fields can be used to count assets.

id (Integer)	type (Keyword: UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN)
name (Text)	tagName (Text)
created (Date)	tagId (Text)
updated (Date)	solrQuery (Text, an asset solr query)

#### Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN

### Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Asset Management Permission "Read Asset"

### Example

#### Request:

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

file.xml

Request POST data (file.xml):

```
<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/xs
d/2.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

## Delete Asset

Delete one or more assets.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/delete/am/asset/<id> or https://<baseurl>/qps/rest/2.0/delete/am/asset
<b>Method</b>	POST

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

Managers with Full Scope

Users without Full Scope must have these account settings: 1) scope includes the requested asset(s), and 2) permissions include:

- Access Permission "API Access"
- Asset Management Permission "Delete Asset"

## Examples

Delete assets with a particular tag.

### Request:

```
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
```

### Request POST data (file.xml):

```
<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/xs  
d/2.0/am/asset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Asset>  
      <id>1972521</id>  
    </Asset>  
  </data>  
</ServiceResponse>
```

## Activate Asset

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) application module and/or the Policy Compliance (PC) module.

<b>URL for VM assets</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/activate/am/asset/&lt;id&gt;?module=QWEB_VM</code> or <code>https://&lt;baseurl&gt;/qps/rest/2.0/activate/am/asset?module=QWEB_VM</code>
<b>URL for PC assets</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/activate/am/asset/&lt;id&gt;?module=QWEB_PC</code> or <code>https://&lt;baseurl&gt;/qps/rest/2.0/activate/am/asset?module=QWEB_PC</code>
<b>Method</b>	POST

## Searchable Fields

These fields can be used to search for assets.

id (Integer)	type (Keyword: UNKOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN)
name (Text)	tagName (Text)
created (Date)	tagId (Text)
updated (Date)	solrQuery (Text, an asset solr query)

### Allowed Operators

Integer	EQUALS, NOT EQUALS, GREATER, LESSER, IN
Text	CONTAINS, EQUALS, NOT EQUALS
Date	EQUALS, NOT EQUALS, GREATER, LESSER
Keyword	EQUALS, NOT EQUALS, IN



## Permissions

Managers with Full Scope

Users without Full Scope must have these account settings: 1) scope includes the requested asset(s), and 2) permissions include:

- Access Permission “API Access”

For PC assets, the Policy Compliance (PC) module must be enabled

## Example

Activate assets with the tag “Export to VM”.

### 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
```

### Request POST data (file.xml):

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

## Asset Fields

Name	Description
name	id (long)
tags (string)	created (date)
	modified (date)
	type (Asset type: UNKNOWN, HOST, SCANNER, WEBAPP, MALWARE_DOMAIN)
	sourceInfo (AssetSourceQList)

## 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.

### TagSimple

id	long (tag primary key)
name	string (tag name)

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.

### Ec2AssetSourceSimple

availabilityZone (string)	imageID (string)
privateDnsName (string)	publicIpAddress (string)
publicDnsName (string)	privateIpAddress (string)
instanceID (string)	monitoringEnabled (boolean)
instanceType (string)	instanceState (AssetSourceStateCode: PENDING, RUNNING, SHUTTING_DOWN, TERMINATED, STOPPING, STOPPED, UNSUPPORTED)

## Host Instance Vulnerability API

The Host Instance Vulnerability API provides a suite of API functions for managing vulnerability instances found on host assets. The supported Host Instance Vulnerability operations are get, count and search.

### Operations

[Get Vulnerability Info](#)

[Search Vulnerabilities](#)

[Count Vulnerabilities](#)

### Reference

[Host Instance Vulnerability Fields](#)

## Get Vulnerability Info

Returns a single host instance vulnerability data by ID. [See Vulnerability Fields](#)

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/get/am/hostinstancevuln/&lt;id&gt;</code>
<b>Method</b>	GET
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for the host instance vulnerability. <a href="#">Learn more</a>

## Permissions

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”
- Asset Management Permission “Read Asset”

## Example

Fetch the host instance vulnerability with the ID 12345.

### 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"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/2.0/am/hostinstancevuln.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <HostInstanceVuln>  
      <id>9534081</id>  
      <hostAssetId>1543621</hostAssetId>
```

## Chapter 5 — Host Instance Vulnerability API

### Get Vulnerability Info

```
<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>
```

## Search Vulnerabilities

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).

<b>URL</b>	https://<baseurl>/qps/rest/2.0/search/am/hostinstancevuln
<b>Method</b>	POST
<b>Pagination</b>	A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for each vulnerability. <a href="#">Learn more</a>

## Searchable Fields

These fields can be used to search for host instance vulnerabilities. [See Vulnerability Fields](#)

id (Integer)	ignored (Boolean)
hostAssetId (Integer)	disabled (Boolean)
qid (Integer)	firstFound (Date)
fqdn (string)	lastFound (Date)
port (Integer)	lastScanned (Date)
ssl (Boolean)	updated (Date)
found (Boolean)	protocol (Protocol: TCP, UDP or ICMP)
	source (Source: HOST, ORACLE or HSSQL, OTHER)

### Allowed Operators

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

## Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Asset Management Permission "Read Asset"

## Example

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

### Request:

```
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
```

### Request POST data (file.xml):

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

## Count Vulnerabilities

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).

<b>URL</b>	https://<baseurl>/qps/rest/2.0/count/am/hostinstancevuln
<b>Method</b>	POST

## Permissions

Managers with Full Scope

Users without Full Scope must have these account permissions:

- Access Permission "API Access"
- Asset Management Permission "Read Asset"

## Example

Count the number of host instance vulnerabilities across all visible assets.

### 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.

```
<ServiceRequest>  
  <filters>  
    <Criteria field="found" operator="EQUALS">true</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/hostinstancevuln.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
</ServiceResponse>
```

## Host Instance Vulnerability Fields

id (long)	The primary host instance vulnerability key.
hostAssetId (long)	The ID of the host asset where the vulnerability was found.
created (date)	The date the vulnerability was added to the KnowledgeBase.
found (boolean)	Set to true if the QID was detected on the host by the latest scan of that host.
firstFound (date)	The date/time the vulnerability was first detected on the host.
lastfound (date)	The most recent date/time the vulnerability was detected on the host.
lastScanned (date)	The most recent date/time the vulnerability was tested for the host.
qid (long)	The Qualys vulnerability ID of the vulnerability.
disabled (long)	Set to true if the QID is marked as disabled in your subscription. Set to false if the QID is not marked disabled.
fqdn (string)	The fully qualified domain name of the host.
ssl (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.
updated (date)	The last date/time the vulnerability data was updated for the host.
ignored (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.
protocol (Protocol: TCP, UDP, ICMP)	The protocol the vulnerability was detected on.
port (integer)	The port number the vulnerability was detected on.
source (VulnSource: HOST, ORACLE, HSSQL, OTHER)	The vulnerability source.

## Asset Data Connector API

The Asset Data Connector API provides a suite of API functions for managing Qualys asset data connectors.

### **Operations**

[Get Connector Info](#)

[Update Connector](#)

[Search Connectors](#)

[Count Connectors](#)

[Delete Connector](#)

[Run Connector](#)

### **Reference**

[Connector Fields](#)

# Get Connector Info

Returns a single asset data connector by ID.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/get/am/assetdataconnector/<id>
<b>Method</b>	GET
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for the asset data connector. <a href="#">Learn more</a>

## Permissions

Managers with Full Scope

## Example

Fetch the asset data connector with the ID 12345.

### 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>
      <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>
```

# Update Connector

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

<b>URL</b>	https://<baseurl>/qps/rest/2.0/update/am/assetdataconnector/ <id> or https://<baseurl>/qps/rest/2.0/update/am/assetdataconnector
<b>Method</b>	POST

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

Managers with Full Scope

## Example

### Request 1:

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

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

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

### Request POST data (file.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/xs
d/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>
```



**Request 2:**

Add a tag to all asset data connectors who's names contains External.

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

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

**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>
```

**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/xs
d/2.0/am/assetdataconnector.xsd">
  <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>
<activation>
  <ActivationModule>VM</ActivationModule>
</activation>
</AssetDataConnector>
...
</data>
</ServiceResponse>
```

# Search Connectors

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

<b>URL</b>	https://<baseurl>/qps/rest/2.0/search/am/assetdataconnector
<b>Method</b>	POST
<b>Pagination</b>	A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for each asset data connector. <a href="#">Learn more</a>

## Searchable Fields

These fields can be used to search for asset data connectors.

id (Integer)	Primary key
name (Text)	
description (Text)	
lastSync (Date)	
lastError (Text)	
connectorState (Keyword)	PENDING, RUNNING, SUCCESS, ERROR
activation (Keyword)	VM or PC
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.

### Allowed Operators

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

## Permissions

Managers with Full Scope

## Example

Find all asset data connectors with tag name USA.

### Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/assetdataconn
ector" < 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"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/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>
<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>
```

## Count Connectors

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

<b>URL</b>	https://<baseurl>/qps/rest/2.0/count/am/assetdataconnector
<b>Method</b>	POST

## Permissions

Managers with Full Scope

## Example

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

### Request:

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

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

### 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"?>  
<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>  
</ServiceResponse>
```

# Delete Connector

Delete one or more asset data connectors.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/delete/am/assetdataconnector/&lt;id&gt;</code> or <code>https://&lt;baseurl&gt;/qps/rest/2.0/delete/am/assetdataconnector</code>
<b>Method</b>	POST

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

Managers with Full Scope

## Examples

### Request 1:

Delete a single asset data connector.

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/2.0/delete/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>  
  <data>  
    <AssetDataConnector>  
      <id>12345</id>  
    </AssetDataConnector>  
  </data>
```

```
</ServiceResponse>
```



Request 2:

Delete several asset data connectors tagged with the To Delete tag.

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

Request POST data (file.xml):

```
<?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/xs
d/2.0/am/assetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AssetDataConnector>
      <id>1972521</id>
    </AssetDataConnector>
  </data>
</ServiceResponse>
```

## Run Connector

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.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/run/am/assetdataconnector/&lt;id&gt;</code> or <code>https://&lt;baseurl&gt;/qps/rest/2.0/run/am/assetdataconnector</code> or
<b>Method</b>	POST

## Permissions

Managers with Full Scope

## Examples

### Request 1:

Run a single connector.

```
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>
  <list>
    <TagSimple>
      <id>3</id>
      <name>USA</name>
    </TagSimple>
  </list>
</defaultTags>
<activation>
  <ActivationModule>VM</ActivationModule>
  <ActivationModule>PC</ActivationModule>
</activation>
</AssetDataConnector>
</data>
</ServiceResponse>

```

**Request 2:**

Re-run all errored connectors.

```

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

```

**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"?>
<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>13</count>
<hasMoreRecords>>false</hasMoreRecords>
<data>
  <AssetDataConnector>
    <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>
  ...

```

## Connector Fields

Name (type)	Description	Writable
id (long)	Primary key	No
name (string)		Yes
description (string)		Yes
lastSynch (date)		No
lastError (string)		No
connectorState (AssetDataConnectorState)	PENDING, RUNNING, SUCCESS, ERROR	No
type (AssetDataConnectorType)	AWS	No
defaultTags (TagSimpleQList)	Tags applied to any asset discovered by the connector	Yes
activation (List<ActivationModule>)	Assets discovered by the connector will be activated for the modules specified	Yes
disabled (boolean)	Whether execution of the connector is disabled (YES). If disabled the connector will not synchronize assets.	Yes

### 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.

#### TagSimple

id	long (tag primary key)
name	string (tag name)

## AWS Asset Data Connector API

The AWS Asset Data Connector API provides a suite of API functions for managing AWS asset data connectors, used to scan EC2 instances.

### Operations

[Get AWS Connector Info](#)

[Create AWS Connector](#)

[Support for AWS GovCloud](#)

[Update AWS Connector](#)

[Search AWS Connectors](#)

[Count AWS Connectors](#)

[Delete AWS Connector](#)

[Run Connector](#)

### Reference

[AWS Connector Fields](#)

# Get AWS Connector Info

Returns a single AWS connector by ID.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/get/am/awsassetdataconnector/<id>
<b>Method</b>	GET
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for the AWS connector. <a href="#">Learn more</a>

## Permissions

Managers with Full Scope

## Examples

Fetch the asset data connector with the ID 12345.

### 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"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/2.0/am/awsassetdataconnector.xsd">  
  <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>  
      <connectorState>ERROR</connectorState>  
      <type>AWS</type>
```

```
<defaultTags>
  <list>
    <TagSimple>
      <id>1</id>
      <name>EC2</name>
    </TagSimple>
  </list>
</defaultTags>
<activation>
  <ActivationModule>VM</ActivationModule>
</activation>
<authRecord>
  <id>1</id>
  <name>my ec2</name>
</authRecord>
<endpoints>
  <list/>
</endpoints>
<allRegions>true</allRegions>
</AwsAssetDataConnector>
</data>
</ServiceResponse>
```

## 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/awsassetdataconn
ector/19201"
```

### 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>
    <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>
    <authRecord1/>
    <endpoints>
      <list>
        <AwsEndpointSimple>
          <regionCode>ap-south-1</regionCode>
        </AwsEndpointSimple>
      </list>
    </endpoints>
    <allRegions>>false</allRegions>
  </AwsAssetDataConnector>
</data>
</ServiceResponse>
```

# Create AWS Connector

Creates an AWS asset data connector.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/create/am/awsassetdataconnector</code>
<b>Method</b>	POST

**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

Managers with Full Scope

## Example

### Example 1: Create new AWS asset data connector

#### 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 the request POST data.

#### Request POST data (file.xml):

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

```

        <id>1</id>
      </TagSimple>
    </set>
  </defaultTags>
  <activation>
    <set>
      <ActivationModule>VM</ActivationModule>
    </set>
  </activation>
  <authRecord>
    <id>1</id>
  </authRecord>
  <allRegions>true</allRegions>
</AwsAssetDataConnector>
</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/xs
d/2.0/am/awssassetdataconnector.xsd">
  <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>
<disabled>>false</disabled>
<isGovCloudConfigured>>false</isGovCloudConfigured>
<authRecord>
  <id>1</id>
  <name>my ec2</name>
</authRecord>
<endpoints>
  <list/>
</endpoints>
<allRegions>>true</allRegions>
</AwsAssetDataConnector>
</data>
</ServiceResponse>
```

### **Example 2: Create new AWS asset data connector in disabled state**

#### Request:

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

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

#### 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>

```

**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/xs
d/2.0/am/awsassetdataconnector.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <AwsAssetDataConnector>
            <id>254401</id>
            <name>disabled-connector</name>
            <connectorState>DISABLED</connectorState>
            <type>AWS</type>
            <defaultTags>
                <list/>
            </defaultTags>
            <activation>
                <list>
                    <ActivationModule>VM</ActivationModule>
                    <ActivationModule>PC</ActivationModule>
                </list>
            </activation>
            <disabled>>true</disabled>
            <isGovCloudConfigured>>false</isGovCloudConfigured>
            <authRecord/>
            <endpoints>
                <list>
                    <AwsEndpointSimple>
                        <regionCode>us-west-1</regionCode>
                    </AwsEndpointSimple>
                    <AwsEndpointSimple>
                        <regionCode>ap-northeast-1</regionCode>
                    </AwsEndpointSimple>
                </list>
            </endpoints>
        </AwsAssetDataConnector>
    </data>
</ServiceResponse>

```

```
        <AwsEndpointSimple>
            <regionCode>eu-west-1</regionCode>
        </AwsEndpointSimple>
    ...
        <AwsEndpointSimple>
            <regionCode>us-east-2</regionCode>
        </AwsEndpointSimple>
    </list>
</endpoints>
<allRegions>true</allRegions>
</AwsAssetDataConnector>
</data>
</ServiceResponse>
```

# Support for AWS GovCloud

**isGovCloudConfigured** (boolean) is used to enable the GovCloud (US) region for an EC2 connector. This parameter can be set for a “create” or “update” request, and is valid only when the GovCloud (US) option is enabled for your subscription.

When **isGovCloudConfigure** is set to “true”

- The connector is configured to pull instance info from the GovCloud (US) region only.
- The connector can't be configured with **allRegions** set to “true”.

## API Request:

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

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

## Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>gov-cloud</name>
      <activation>
        <set>
          <ActivationModule>VM</ActivationModule>
          <ActivationModule>PC</ActivationModule>
        </set>
      </activation>
      <isGovCloudConfigured>true</isGovCloudConfigured>
      <authRecord>
        <id>134601</id>
      </authRecord>
      <allRegions>false</allRegions>
      <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="https://qualysapi.qualys.com/qps/xs
d/2.0/am/awsassetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>149008</id>
      <name>gov-cloud</name>
      <connectorState>PENDING</connectorState>
      <type>AWS</type>
      <defaultTags>
        <list/>
      </defaultTags>
      <activation>
        <list>
          <ActivationModule>VM</ActivationModule>
          <ActivationModule>PC</ActivationModule>
        </list>
      </activation>
      <disabled>>false</disabled>
      <isGovCloudConfigured>>true</isGovCloudConfigured>
      <authRecord/>
      <endpoints>
        <list>
          <AwsEndpointSimple>
            <regionCode>us-gov-west-1</regionCode>
          </AwsEndpointSimple>
        </list>
      </endpoints>
      <allRegions>>false</allRegions>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
```



## Support 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”.

### Example

#### API request:

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

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

#### Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>cn-north-1-connl</name>
      <authRecord>
        <id>132601</id>
      </authRecord>
      <endpoints>
        <add>
          <AwsEndpointSimple>
            <regionCode>cn-north-1</regionCode>
          </AwsEndpointSimple>
        </add>
      </endpoints>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>
```

```
        <isChinaConfigured>true</isChinaConfigured>
        <disabled>>false</disabled>
    </AwsAssetDataConnector>
</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/xs
d/2.0/am/aws_asset_data_connector.xsd">
    <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>
            <allRegions>>false</allRegions>
```

```
</AwsAssetDataConnector>  
</data>  
</ServiceResponse>
```

## Support for 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.

### Examples

#### Create new connector

Use this method to create a new connector when you already have the ARN generated from your AWS account.

##### 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/awsassetdatac
onnector" < file.xml
```

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>
      <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>
      <qualysAwsAccountId>383031258652</qualysAwsAccountId>
      <authRecord/>
      <endpoints>
        <list>
          <AwsEndpointSimple>
            <regionCode>ap-south-1</regionCode>
          </AwsEndpointSimple>
        </list>
      </endpoints>
      <allRegions>>false</allRegions>
```

```
    </AwsAssetDataConnector>  
  </data>  
</ServiceResponse>
```

## Create new connector (provide 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:

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

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>  
      </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>19201</id>
    <name>my-aws-connector</name>
    <connectorState>INCOMPLETE</connectorState>
    <type>AWS</type>
    <defaultTags>
      <list/>
    </defaultTags>
    <activation>
      <list/>
    </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>
```

# Update AWS Connector

Updates writable fields and collections.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/update/am/awsassetdataconnector/&lt;id&gt;</code> or <code>https://&lt;baseurl&gt;/qps/rest/2.0/update/am/awsassetdataconnector</code>
<b>Method</b>	POST

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

Managers with Full Scope

## Examples

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.

### Request:

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

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

### Request POST data (file.xml):

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



```

        </TagSimple>
    </add>
</defaultTags>
<endpoints>
    <add>
        <AwsEndpointSimple>
            <regionCode>us-east-1</regionCode>
        </AwsEndpointSimple>
    </add>
</endpoints>
</AwsAssetDataConnector>
</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/awssassetdataconnector.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>

```

```
</activation>
<disabled>>false</disabled>
<isGovCloudConfigured>>false</isGovCloudConfigured>
</AssetDataConnector>
</data>
</ServiceResponse>
```

## 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/awsassetdatac
onnector/12345" < file.xml
```

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

### Request POST data (file.xml):

```
<?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>
```

## Search AWS Connectors

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

<b>URL</b>	https://<baseurl>/qps/rest/2.0/search/am/awsassetdataconnector
<b>Method</b>	POST
<b>Pagination</b>	A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for each AWS connector. <a href="#">Learn more</a>

### Searchable Fields

These fields can be used to search for AWS connectors.

id (Integer)	Primary key
name (Text)	
description (Text)	
lastSync (Date)	
lastError (Text)	
connectorState (Keyword)	PENDING, RUNNING, SUCCESS, ERROR
activation (Keyword)	VM or PC
defaultTags.name (Text)	The name of a tag in the defaultTags collection
defaultTag (Integer)	The ID of a tag in the defaultTags collection
allRegions (Boolean)	
serviceType (Keyword)	EC2
endpoint.region (Text)	AWS region code
authRecord (Integer)	The ID of the authentication record

authRecord.name (Text)	The name of the authentication record
disabled (Boolean)	Whether execution of the connector is disabled (YES). If disabled the connector will not synchronize assets.

Allowed Operators

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

## Permissions

Managers with Full Scope

## Example

Find all asset data connectors with tag name USA.

Request:

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

## Count AWS Connectors

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

<b>URL</b>	https://<baseurl>/qps/rest/2.0/count/am/awsassetdataconnector
<b>Method</b>	POST

### Permissions

Managers with Full Scope

### Example

Count the number of AWS connectors with the tag name USA.

#### Request:

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

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

#### 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"?>  
<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>
```

# Delete AWS Connector

Delete one or more AWS connectors.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/delete/am/awsassetdataconnector/&lt;id&gt;</code> or <code>https://&lt;baseurl&gt;/qps/rest/2.0/delete/am/awsassetdataconnector</code>
<b>Method</b>	POST

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

Managers with Full Scope

## Examples

### Request 1:

Delete a single AWS connector.

```
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>
```



Request 2:

Delete several AWS connectors tagged with the To Delete tag.

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

Request POST data (file.xml):

```
<?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/xs
d/2.0/am/awsassetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AssetDataConnector>
      <id>1972521</id>
    </AssetDataConnector>
  </data>
</ServiceResponse>
```

## Run Connector

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.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/run/am/assetdataconnect or/&lt;id&gt; or https://&lt;baseurl&gt;/qps/rest/2.0/run/am/assetdataconnect or</code>
<b>Method</b>	POST

### Permissions

Managers with Full Scope

### See also

See also [Run Connector](#)

# AWS Connector Fields

Name (type)	Description	Writable
id (long)	Primary key	No
name (string)		Yes
description (string)		Yes
lastSynch (date)		No
lastError (string)		No
connectorState (AssetDataConnectorState)	PENDING, RUNNING, SUCCESS, ERROR, DISABLED, INCOMPLETE	No
type (AssetDataConnectorType)	AWS	No
defaultTags (TagSimpleQList)	Tags applied to any asset discovered by the connector	Yes
activation (List<ActivationModule>)	Assets discovered by the connector will be activated for the modules specified	Yes
authRecord (AwsAuthRecordSimple)	The AWS authentication record the connector will use to connect to AWS. When writing/updating it is looked up by the ID field.	Yes
serviceType (AwsServiceType)	EC2	Yes
allRegions (boolean)	If true the end point's collection will be ignored an all AWS regions scanned	Yes
disabled (boolean)	Whether execution of the connector is disabled (YES). If disabled the connector will not synchronize assets.	Yes
arn	Generated by AWS. Ensure that you provide the same ARN that is generated by AWS.	yes
externalId	Random string which is unique for each user.	yes

## 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 API

The AWS Authentication Record API provides a suite of API functions for managing AWS authentication records, used to scan EC2 instances.

### Operations

[Get AWS Auth Record Info](#)

[Create AWS Auth Record](#)

[Update AWS Auth Record](#)

[Search AWS Auth Records](#)

[Count AWS Auth Records](#)

[Delete AWS Auth Record](#)

### Reference

[AWS Auth Record Fields](#)

## Get AWS Auth Record Info

Returns a single AWS authentication record by ID.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/get/am/awsauthrecord/&lt;id&gt;</code>
<b>Method</b>	GET
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for the authentication record. <a href="#">Learn more</a>

### Permissions

Managers with Full Scope

### Example

Fetch the authentication record with the ID 12345.

Request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/2.0/get/am/awsauthrecord/12  
345"
```

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/xs  
d/2.0/am/awsauthrecord.xsd">  
  <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>
```

# Create AWS Auth Record

Creates a new authentication record.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/create/am/awsauthrecord
<b>Method</b>	POST

## Permissions

Managers with Full Scope

## Example

Create a new authentication record.

### 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 the request POST data.

### Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAuthRecord>
      <name>Simple Auth Record</name>
      <description>Production Auth Record</description>
      <accessKeyId>AAAAAAAAAAAAAAAAA11A</accessKeyId>
<secretKey>1a1Aa1aaaaa1aAaAaaAaa1Aaaaa11aaAAAAaaaA</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/xs
```

```
d/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>
```



# Update AWS Auth Record

Updates writable fields.

<b>URL</b>	https://<baseurl>/qps/rest/2.0/update/am/awsauthrecord/<id> or https://<baseurl>/qps/rest/2.0/update/am/awsauthrecord
<b>Method</b>	POST

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

Managers with Full Scope

## Example

Update the secret key of auth record 12345

### Request:

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

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

### Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAuthRecord>
<secretKey>1a1laa1aaaaa1aAaAaaAaa1Aaaaa11aaAAAAaaaA</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/xs
d/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>
```

# Search AWS Auth Records

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

<b>URL</b>	https://<baseurl>/qps/rest/2.0/search/am/awsauthrecord
<b>Method</b>	POST
<b>Pagination</b>	A maximum of 100 instances are returned by default. To customize this specify a “preferences” tag in the POST body of your request.
<b>Limit your results</b>	Use the optional “fields” parameter to limit the amount of information returned for each authentication record. <a href="#">Learn more</a>

## Searchable Fields

These fields can be used to search for authentication records.

---

id (Integer)

---

name (Text)

---

description (Text)

---

created (Date)

---

modified (Date)

Allowed Operators

Integer      EQUALS, NOT EQUALS,  
                 GREATER, LESSER, IN

Text            CONTAINS, EQUALS,  
                 NOT EQUALS

Date            EQUALS, NOT EQUALS,  
                 GREATER, LESSER

## Permissions

Managers with Full Scope

## Example

Find all authentication records that have a name that contains the string AUTH.

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
```

Request POST data (file.xml):

```
<?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>  
      <name>Simple Auth Record 4</name>  
      <modified>2014-12-22T19:11:18Z</modified>
```

```
    </AwsAuthRecord>  
  </data>  
</ServiceResponse>
```

## Count AWS Auth Records

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

**URL** `https://<baseurl>/qps/rest/2.0/count/am/awsauthrecord`  
**Method** `POST`

### Permissions

Managers with Full Scope

### Example

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

Request:

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

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

Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>  
<ServiceRequest>  
  <filters>  
    <Criteria field="name operator="CONTAINS">AUTH</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/xs  
d/2.0/am/awsauthrecord.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
</ServiceResponse>
```

# Delete AWS Auth Record

Delete one or more authentication records.

<b>URL</b>	<code>https://&lt;baseurl&gt;/qps/rest/2.0/delete/am/awsauthrecord/&lt;id&gt;</code> or <code>https://&lt;baseurl&gt;/qps/rest/2.0/delete/am/awsauthrecord</code>
<b>Method</b>	POST

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

Managers with Full Scope

## Examples

### Request 1:

Delete a single authentication record.

```
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"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/2.0/am/awsauthrecord.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <AssetDataConnector>  
      <id>12345</id>  
    </AssetDataConnector>  
  </data>
```

```
</ServiceResponse>
```



Request 2:

Delete several authentication records whose names contain the string "delete me".

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

Request POST data (file.xml):

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">delete
me</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/xs
d/2.0/am/awsauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAuthRecord>
      <id>2020094</id>
    </AwsAuthRecord>
  </data>
</ServiceResponse>
```

## AWS Auth Record Fields

<b>Name (type)</b>	<b>Description</b>	<b>Writable</b>
name (string)	Name of the authentication record	Yes
description (string)	Brief description of the authentication record	Yes
created (date)	When record was created	No
modified (date)	When records was last modified	No
secretKey (string)	The AWS secret key - write only, cannot be read	Yes
accessKeyId (string)	The AWS access key - write only, cannot be read	Yes

## JSON Support

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

### Headers used in samples

Send JSON request	"Content-Type: application/json"
Get response in JSON	"Accept: application/json"

### Example 1: Create a tag

#### Request:

```
cat createTag.json | curl -s -k -X POST -H "Accept: application/json" -  
H "Content-Type: application/json" -H "user: acme_ss2" -H "password:  
passwd" -d @-  
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/tag"
```

#### POST data:

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

```

    }
  }
}
]

```

Response:

```

{
  "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"
}

```

**Example 2: Search tags**Request:

```
cat searchTag.json | curl -s -k -X POST -H "Accept: application/json" -H
"Content-Type: application/json" -H "user: acme_ss2" -H "password: passwd"
-d @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/tag"
```

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"
      }
    ]
  }
}
```

Response:

```
{
  "ServiceResponse" : {
    "data" : [ {
```

## Appendix A — JSON Support

```
"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"
}
```