



# Qualys Cloud Platform v2.x

## API Release Notes

Version 2.33

June 25, 2018

Qualys Cloud Suite API gives you many ways to integrate your programs and API calls with Qualys capabilities. You'll find all the details in our user guides, available at the time of release. Just log in to your Qualys account and go to Help > Resources.

### What's New

[Schedule auto-update for appliances registered to a cluster](#)

[Validate XML/JSON payload](#)

[Easily identify ignored WAS findings](#)

[Uninstall Cloud Agent using UUID](#)

[HostAsset and Asset APIs show new Cloud Provider metadata for AWS, Azure and GCP](#)

### 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.

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

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.

## Schedule auto-update for appliances registered to a cluster

|                     |                                                                      |
|---------------------|----------------------------------------------------------------------|
| API affected        | /qps/rest/2.0/create/waf/cluster<br>/qps/rest/2.0/update/waf/cluster |
| New or Updated APIs | Updated                                                              |
| DTD or XSD changes  | Yes                                                                  |

You can now use the cluster API to specify when the appliances registered with a cluster get auto-updated. Specify days of the week and the start time. By default, auto-update is enabled for all days of the week.

You can choose to freeze auto-updates until a specific date. Auto-updates are stopped up to the end date and then resumed.

### Input Parameters

New input parameters are described below.

| Parameter                      | Description                                                                                                                   |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| updateSchedule.enabled         | (Boolean) Schedule Auto-Update enabled or disabled. By default this is enabled.                                               |
| updateSchedule.weekDays        | (Text) Comma separated list of days the auto-update is allowed. By default enabled for all days: MON,TUE,WED,THU,FRI,SAT,SUN. |
| updateSchedule.startTime       | (Integer) Hour of the day the auto-update may occur. Default is 0. Specify the hour in 24 hour format.                        |
| updateSchedule.timezone.code   | (Text) The timezone. For example, Pacific Standard Time. Default is UTC.                                                      |
| updateSchedule.timezone.offset | (Text) The timezone offset (hours and minutes). Default is +00:00.                                                            |
| updateSchedule.freezeEndDate   | (Date) If set the auto-update process will not be executed before the specified date (e.g. 2018-03-01). Disabled by default.  |

### Example

Here's sample request and output showing scheduled auto-update.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"http://qualysapi.qualys.com/qps/rest/2.0/update/waf/cluster/210601" <
file.xml
```

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

Request POST data: (Contents of file.xml)

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <Cluster>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
          <code>Asia/Aden</code>
          <offset>+03:00</offset>
        </timezone>
        <freezeEndDate>2018-02-28</freezeEndDate>
      </updateSchedule>
    </Cluster>
  </data>
</ServiceRequest>
```

XML output:

```
<?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/wa
f/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>210601</id>
      <uuid>f43cf526-7a52-4fb5-ad4c-6d118ded6329</uuid>
      <name>
        <![CDATA[TestClusterfromAddddPI123]]>
      </name>
      <errorResponse>
        <block/>
      </errorResponse>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
          <code>Asia/Aden</code>
          <offset>+03:00</offset>
        </timezone>
        <nextUpgradeDate>2018-05-14T23:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
```

```
        <id>11826614</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
    </owner>
    <created>2018-05-09T23:21:52Z</created>
    <createdBy>
        <id>11826614</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
    </createdBy>
    <updated>2018-05-11T09:20:27Z</updated>
    <updatedBy>
        <id>11826614</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
    </updatedBy>
    <token>
        <![CDATA[AECDFB82-F704-4E40-9C42-2ED207F0CDF2]]>
    </token>
    <syncDate>2018-05-11T08:52:17Z</syncDate>
    <status>NO_SENSORS</status>
    <deploymentStatus>UNUSED</deploymentStatus>
    <deployed>2018-05-09T23:21:52Z</deployed>
</Cluster>
</data>
</ServiceResponse>
```

XSD update:

Changes in Clusters XSD (/qps/xsd/2.0/waf/cluster.xsd)

```
...
<xs:complexType name="TimeZone">
    <xs:sequence>
        <xs:element name="code" type="xs:string" minOccurs="0"/>
        <xs:element name="offset" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="LocalDate">
    <xs:restriction base="xs:string">
        <xs:pattern value="\d\d\d\d-\d\d-\d\d" />
    </xs:restriction>
</xs:simpleType>

...
```

```
<xs:complexType name="UpdateSchedule">
  <xs:sequence>
    <xs:element name="enabled" type="xs:boolean" minOccurs="1"
maxOccurs="1" />
    <xs:element name="weekDays" type="xs:string" minOccurs="0"
maxOccurs="1" />
    <xs:element name="startTime" minOccurs="0" maxOccurs="1" >
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="23"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="timezone" type="TimeZone" minOccurs="0"
maxOccurs="1" />
    <xs:element name="freezeEndDate" type="LocalDate"
minOccurs="0" maxOccurs="1" />
    <xs:element name="nextUpgradeDate" type="xs:dateTime"
minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

...

    <xs:element name="updateSchedule" type="UpdateSchedule"
minOccurs="0"/>
  ...
```

## Validate XML/JSON payload

|                     |                                                                              |
|---------------------|------------------------------------------------------------------------------|
| API affected        | /qps/rest/2.0/create/waf/httpprofile<br>/qps/rest/2.0/update/waf/httpprofile |
| New or Updated APIs | Updated                                                                      |
| DTD or XSD changes  | Yes                                                                          |

You can now use the HTTP Profiles API to enable XML/JSON parsing to validate that transmitted payload is XML/JSON compliant. Parsing is not enabled by default.

## Input Parameters

New input parameters are described below.

| Parameter                                    | Description                                                                                                                                                                                                                                                                                                                                |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| webServiceProtection.xmlP<br>arsing.enabled  | (Boolean) XML parser enabled or disabled. Default is disabled.                                                                                                                                                                                                                                                                             |
| webServiceProtection.xmlP<br>arsing.size     | (Integer) Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters. |
| webServiceProtection.xmlP<br>arsing.items    | (Integer) Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.                                                                                                                                                                                   |
| webServiceProtection.xmlP<br>arsing.level    | (Integer) Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.                                                                                                                                                       |
| webServiceProtection.jsonP<br>arsing.enabled | (Boolean) JSON parser enabled or disabled. Default is disabled.                                                                                                                                                                                                                                                                            |
| webServiceProtection.jsonP<br>arsing.size    | (Integer) Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters. |
| webServiceProtection.jsonP<br>arsing.items   | (Integer) Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.                                                                                                                                                                                   |

| Parameter                                  | Description                                                                                                                                                                          |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| webServiceProtection.jsonP<br>arsing.level | (Integer) Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32. |

## Example

Here's sample request and output showing XML/JSON parsing.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/update/waf/httpprofile/110601"
" <
file.xml
```

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

Request POST data: (Contents of file.xml)

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <HTTPProfile>
      <webServiceProtection>
        <xmlParsing>
          <enabled>true</enabled>
          <size>500000</size>
          <items>50000</items>
          <level>64</level>
        </xmlParsing>
        <jsonParsing>
          <enabled>false</enabled>
        </jsonParsing>
      </webServiceProtection>
    </HTTPProfile>
  </data>
</ServiceRequest>
```

XML output:

```
<?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/wa
f/httpprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HTTPProfile>
```

```

<id>110601</id>
<uuid>a5550176-0675-483c-80aa-7859ee1860c3</uuid>
<name>
  <![CDATA[11Copy of requestContentTypewithDenyAllww123]]>
</name>
<owner>
  <id>11826614</id>
  <username>john_doe</username>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
</owner>
<created>2018-05-09T21:23:01Z</created>
<createdBy>
  <id>11826614</id>
  <username>john_doe</username>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
</createdBy>
<updated>2018-05-11T09:28:23Z</updated>
<updatedBy>
  <id>11826614</id>
  <username>john_doe</username>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
</updatedBy>
<system>false</system>
<requestMethod>
  <denyAll>
    <![CDATA[CONNECT, COPY, DELETE, GET, HEAD, LOCK, MKCOL,
MOVE, OPTIONS, POST, PROPFIND, PROPPATCH, PUT, TRACE, TRACK, UNLOCK]]>
  </denyAll>
</requestMethod>
<requestHeader>
  <detectInvalid>true</detectInvalid>
  <detectRepeated>true</detectRepeated>
  <detectChunked>true</detectChunked>
</requestHeader>
<requestContentType>
  <denyAll>
    <![CDATA[application/x-www-form-urlencoded
multipart/form-data
text/xml
application/xml
application/x-amf]]>
  </denyAll>
</requestContentType>
<detectProtocolAnomalies>true</detectProtocolAnomalies>
<webServiceProtection>
  <xmlParsing>

```

```

        <enabled>true</enabled>
        <size>500000</size>
        <items>50000</items>
        <level>64</level>
    </xmlParsing>
    <jsonParsing>
        <enabled>false</enabled>
    </jsonParsing>
</webServiceProtection>
<serverCloaking>
    <enabled>true</enabled>
    <value>
        <![CDATA[tomcat]]>
    </value>
</serverCloaking>
<suppressSensitiveHeaders>true</suppressSensitiveHeaders>
<onErrorMessages>ALLOW</onErrorMessages>
<onSensitiveFileTypes>LOG</onSensitiveFileTypes>
<onSensitiveFileExtensions>LOG</onSensitiveFileExtensions>
<cookieProtection>
    <type>NONE</type>
</cookieProtection>

<discourageContentTypeSniffing>true</discourageContentTypeSniffing>
    <forceDefaultContentType>
        <enabled>false</enabled>
    </forceDefaultContentType>
    <forceDefaultCharacterEncoding>
        <type>ALWAYS_APPLY</type>
        <value>UTF_8</value>
    </forceDefaultCharacterEncoding>
    <contentSecurityPolicyHeader>
        <enabled>false</enabled>
    </contentSecurityPolicyHeader>
    <discourageClickjacking>NONE</discourageClickjacking>

<browserXSSProtection>ENABLE_WITH_BLOCKING</browserXSSProtection>
</HTTPProfile>
</data>
</ServiceResponse>

```

XSD update:

Changes in Http Profiles XSD (/qps/xsd/2.0/waf/httpprofile.xsd)

```
...
    <xs:simpleType name="PositiveLong">
        <xs:restriction base="xs:long">
            <xs:minInclusive value="1"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="ParsingSettings">
        <xs:sequence>
            <xs:element name="enabled" type="xs:boolean" minOccurs="1"
maxOccurs="1" />
            <xs:element name="size" type="PositiveLong" minOccurs="0"
maxOccurs="1" />
            <xs:element name="items" type="PositiveLong" minOccurs="0"
maxOccurs="1" />
            <xs:element name="level" type="PositiveLong" minOccurs="0"
maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>

...

    <xs:element name="webServiceProtection" minOccurs="0"
maxOccurs="1">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="xmlParsing"
type="ParsingSettings" minOccurs="1" maxOccurs="1"/>
                <xs:element name="jsonParsing"
type="ParsingSettings" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

...
```

## Easily identify ignored WAS findings

|                     |                                  |
|---------------------|----------------------------------|
| API affected        | /qps/rest/3.0/search/was/finding |
| New or Updated APIs | Updated                          |
| DTD or XSD changes  | No                               |

We have introduced a new element `<isIgnored>` to easily identify whether a WAS finding (detection) in the user's scope is ignored or not.

The `<isIgnored>` element will be returned even when you are not taking a verbose output.

### Example

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/finding" <
file.xml
```

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

Request POST data: (Contents of file.xml)

```
<?xml version="1.0" encoding="UTF-8"?><ServiceRequest>
<filters>
<Criteria field="id" operator="EQUALS">321380</Criteria>
</filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Finding>
      <id>321380</id>
      <qid>150004</qid>
      <name>
        <![CDATA[Path-Based Vulnerability]]>
      </name>
      <type>VULNERABILITY</type>
      <findingType>QUALYS</findingType>
```

```
<severity>2</severity>
<url>
  <![CDATA[http://funkytown.vuln.qa..com/cassium/css/.]]>
</url>
<status>NEW</status>
<firstDetectedDate>2017-04-26T09:26:55Z</firstDetectedDate>
<lastDetectedDate>2017-04-26T09:26:55Z</lastDetectedDate>
<lastTestedDate>2017-04-26T09:26:55Z</lastTestedDate>
<timesDetected>1</timesDetected>
<webApp>
  <id>5264856</id>
  <name>
    <![CDATA[CDATA My Web Application1493015338569]]>
  </name>
  <url>
    <![CDATA[http://funkytown.acme01.com:80/cassium/xss/]]>
  </url>
</webApp>
<isIgnored>>false</isIgnored>
</Finding>
</data>
</ServiceResponse>
```

## Uninstall Cloud Agent using UUID

|                     |                                                                          |
|---------------------|--------------------------------------------------------------------------|
| API affected        | /qps/rest/2.0/uninstall/am/asset<br>/qps/rest/2.0/uninstall/am/hostasset |
| New or Updated APIs | Updated                                                                  |
| DTD or XSD changes  | No                                                                       |

Cloud Agent uninstall API now allows you to specify the agent UUID to identify an agent during uninstallation. Agent UUID can be specified in the request XML.

Currently this is supported only for bulk uninstall.

### Example

Here's sample request and output showing the use of agent UUID for uninstalling agents using Asset API.

Agents can be uninstalled using the Host Asset API in the similar way.

API request:

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

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

Request POST data: (Contents of file.xml)

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <filters>
    <Criteria field="agentUuid" operator="IN">1d71c07e-f1b0-42df-918b-
01b231489b86,0285fead-0014-465e-8840-6457ad482bb3</Criteria>
  </filters>
</ServiceRequest>
```

Note: Use the EQUALS operator for uninstalling a single agent or use the IN operator with comma separated UUID values for bulk uninstall.

XML output:

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

```
<data>
  <Asset>
    <id>712038</id>
    <name>192.168.1.12</name>
    <created>2017-06-18T16:54:50Z</created>
    <modified>2018-04-18T16:54:51Z</modified>
    <type>HOST</type>
    <tags>
      <list>
        <TagSimple>
          <id>509525</id>
          <name>Cloud Agent</name>
        </TagSimple>
      </list>
    </tags>
    <sourceInfo>
      <list>
        <AssetSource/>
      </list>
    </sourceInfo>
  </Asset>
  <Asset>
    <id>2532638</id>
    <name>10.40.2.176</name>
    <created>2017-04-17T13:32:48Z</created>
    <modified>2017-05-14T14:06:03Z</modified>
    <type>HOST</type>
    <tags>
      <list>
        <TagSimple>
          <id>509525</id>
          <name>Cloud Agent</name>
        </TagSimple>
      </list>
    </tags>
    <sourceInfo>
      <list>
        <AssetSource/>
      </list>
    </sourceInfo>
  </Asset>
</data>
</ServiceResponse>
```

# HostAsset and Asset APIs show new Cloud Provider metadata for AWS, Azure and GCP

|                     |                                                                |
|---------------------|----------------------------------------------------------------|
| API affected        | /qps/rest/2.0/get/am/hostasset/<br>/qps/rest/2.0/get/am/asset/ |
| New or Updated APIs | Neither (output change only)                                   |
| DTD or XSD changes  | Yes                                                            |

With this release Qualys Cloud Platform shows additional Cloud Provider metadata to users for Amazon AWS, Azure, and Google Cloud Platform. This asset metadata is collected from Vulnerability Scans (using VM), Compliance Scans (using PC or SCA), Cloud Agents and Data Connectors.

- Cloud Provider metadata is visible to Qualys users using the Qualys UI and API
- Asset details UI in Asset View (AV) and Cloud Agent (CA).
  - API results returned by HostAsset and Asset APIs per below. (In previous release, these APIs provided AWS metadata collected from scans)

## Example - Asset API output with Azure output

API request:

```
curl -u "USERNAME:PASSWORD" -X GET
https://qualysapi.qualys.com/qps/rest/2.0/get/am/asset/1841619
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/a
m/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>1841619</id>
      <name>test-azure-auto-1</name>
      <created>2018-04-23T18:04:17Z</created>
      <modified>2018-04-23T18:27:51Z</modified>
      <type>HOST</type>
      <tags>
        <list>
          <TagSimple>
            <id>7511024</id>
            <name>Cloud Agent</name>
          </TagSimple>
        </list>
      </tags>
    </Asset>
  </data>
</ServiceResponse>
```

```
        </list>
      </tags>
      <sourceInfo>
        <list>
          <AssetSource/>
          <AzureAssetSourceSimple>
            <assetId>1841619</assetId>
            <type>AZURE</type>
            <firstDiscovered>2018-04-
23T18:05:04Z</firstDiscovered>
            <lastUpdated>2018-04-23T18:05:05Z</lastUpdated>
            <name>ubuntu162</name>
            <location>SouthIndia</location>
            <vmSize>Standard_A0</vmSize>
            <vmId>AzureVM-1234567</vmId>
            <offer>UbuntuServer</offer>
            <state>RUNNING</state>
            <publisher>Canonical</publisher>
            <version>16.04.201711211</version>
            <osType>Linux</osType>
            <subnet>10.19.41.0</subnet>
            <ipv6>{"&quot;ipAddress&quot;: []}</ipv6>
            <subscriptionId>fbb9ea64-abda-452e-adfa-
83442409e8fe</subscriptionId>
            <resourceGroupName>azureqa</resourceGroupName>
            <publicIpAddress>52.172.37.187</publicIpAddress>
            <privateIpAddress>10.19.41.5</privateIpAddress>
          </AzureAssetSourceSimple>
        </list>
      </sourceInfo>
    </Asset>
  </data>
</ServiceResponse>
```

## Example - Asset API output with GCP output

API request:

```
curl -u "USERNAME:PASSWORD" -X GET
https://qualysapi.qualys.com/qps/rest/2.0/get/am/asset/1841620
```

XML output:

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

```

<data>
  <Asset>
    <id>1841620</id>
    <name>test-gcp-auto-1</name>
    <created>2018-04-23T18:11:05Z</created>
    <modified>2018-04-23T18:11:05Z</modified>
    <type>HOST</type>
    <tags>
      <list>
        <TagSimple>
          <id>7511024</id>
          <name>Cloud Agent</name>
        </TagSimple>
      </list>
    </tags>
    <sourceInfo>
      <list>
        <GcpAssetSourceSimple>
          <assetId>1841620</assetId>
          <type>GCP</type>
          <firstDiscovered>2018-04-
23T18:11:08Z</firstDiscovered>
          <lastUpdated>2018-04-23T18:11:08Z</lastUpdated>
          <instanceId>GCP-1234567</instanceId>
          <hostname>test-gcp</hostname>
          <machineType>test</machineType>
          <zone>useast1</zone>
          <projectIdNo>12334</projectIdNo>
          <state>RUNNING</state>
          <projectId>5677</projectId>
          <network>testNetwork</network>
          <publicIpAddress>34.21.36.11</publicIpAddress>
          <privateIpAddress>10.10.10.10</privateIpAddress>
        </GcpAssetSourceSimple>
        <AssetSource/>
      </list>
    </sourceInfo>
  </Asset>
</data>
</ServiceResponse>

```

## Example - HostAsset API output with Azure metadata info along with tags

API request:

```
curl -u "USERNAME:PASSWORD" -X GET  
https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostasset/1841619
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/a  
m/hostasset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <HostAsset>  
      <id>1841619</id>  
      <name>test-azure-auto-1</name>  
      <created>2018-04-23T18:04:17Z</created>  
      <modified>2018-04-23T18:27:51Z</modified>  
      <type>HOST</type>  
      <tags>  
        <list>  
          <TagSimple>  
            <id>7511024</id>  
            <name>Cloud Agent</name>  
          </TagSimple>  
        </list>  
      </tags>  
      <sourceInfo>  
        <list>  
          <AssetSource/>  
          <AzureAssetSourceSimple>  
            <assetId>1841619</assetId>  
            <type>AZURE</type>  
            <firstDiscovered>2018-04-  
23T18:05:04Z</firstDiscovered>  
            <lastUpdated>2018-04-23T18:05:05Z</lastUpdated>  
            <azureVmTags>  
              <tags>  
                <list>  
                  <AzureTags>  
                    <key>CreatedBy</key>  
                    <value>DK</value>  
                  </AzureTags>  
                  <AzureTags>  
                    <key>project</key>  
                    <value>qU@LY$</value>  
                  </AzureTags>  
                </list>  
              </tags>  
            </azureVmTags>  
          </list>  
        </sourceInfo>  
      </HostAsset>  
    </data>  
  </ServiceResponse>  
</xml>
```

```

        <AzureTags>
          <key>Env</key>
          <value>DevTestLabs</value>
        </AzureTags>
        <AzureTags>
          <key>server-project</key>
          <value>cv360-server-test</value>
        </AzureTags>
        <AzureTags>
          <key>Location</key>
          <value>SOUTHIND</value>
        </AzureTags>
        <AzureTags>
          <key>Name</key>
          <value>AZURE-CENTOS681</value>
        </AzureTags>
      </list>
    </tags>
  </azureVmTags>
  <name>ubuntu162</name>
  <location>SouthIndia</location>
  <vmSize>Standard_A0</vmSize>
  <vmId>AzureVM-1234567</vmId>
  <offer>UbuntuServer</offer>
  <state>RUNNING</state>
  <publisher>Canonical</publisher>
  <version>16.04.201711211</version>
  <osType>Linux</osType>
  <subnet>10.19.41.0</subnet>
  <ipv6></ipv6>
  <subscriptionId>fbb9ea64-abda-452e-adfa-
83442409e8fe</subscriptionId>
  <resourceGroupName>azureqa</resourceGroupName>
  <publicIpAddress>52.172.37.187</publicIpAddress>
  <privateIpAddress>10.19.41.5</privateIpAddress>
  </AzureAssetSourceSimple>
</list>
</sourceInfo>
<lastSystemBoot>2018-04-19T23:52:41Z</lastSystemBoot>
<lastLoggedOnUser>lxagent</lastLoggedOnUser>
<os>Ubuntu Linux 16.04.3</os>

<dnsHostName>ubuntu162.f004wtlrrhz2u3lcrb2sb010z2h.tx.internal.cloudapp.ne
t</dnsHostName>
  <agentInfo>
    <agentVersion>1.7.1.33</agentVersion>
    <agentId>02d31074-5c67-786a-b321-a8c34159daa9</agentId>
    <status>STATUS_INACTIVE</status>
    <lastCheckedIn>2018-04-23T09:05:04Z</lastCheckedIn>
  </agentInfo>

```

```

<connectedFrom>10.11.65.144</connectedFrom>
<chirpStatus>Inventory Scan Complete</chirpStatus>
<platform>Linux</platform>
<activatedModule>NONE</activatedModule>
<agentConfiguration>
  <name>Dev Profile</name>
  <id>1001</id>
</agentConfiguration>
<activationKey>
  <activationId>d4f3c77b-55cd-43ea-8c08-
b9225cccb787</activationId>
  <title>agent-testing</title>
</activationKey>
</agentInfo>
<address>10.11.65.144</address>
<trackingMethod>QAGENT</trackingMethod>
<manufacturer>Microsoft Corporation</manufacturer>
<totalMemory>647</totalMemory>
<timezone>UTC</timezone>
<biosDescription>Virtual Machine, 7.0</biosDescription>
<openPort>
  <list>
    <HostAssetOpenPort>
      <port>68</port>
      <protocol>UDP</protocol>
    </HostAssetOpenPort>
    <HostAssetOpenPort>
      <port>22</port>
      <protocol>TCP</protocol>
    </HostAssetOpenPort>
    <HostAssetOpenPort>
      <port>25324</port>
      <protocol>TCP</protocol>
    </HostAssetOpenPort>
    <HostAssetOpenPort>
      <port>25224</port>
      <protocol>UDP</protocol>
    </HostAssetOpenPort>
  </list>
</openPort>
<software>
  <list>
    <HostAssetSoftware>
      <name>libapparmor-perl</name>
      <version>2.10.95-0ubuntu2.7</version>
    </HostAssetSoftware>
    <HostAssetSoftware>
      <name>ifupdown</name>
      <version>0.8.10ubuntu1.2</version>

```

```

        </HostAssetSoftware>
    </list>
</software>
<processor>
    <list>
        <HostAssetProcessor>
            <name>Intel (R)  Xeon (R) </name>
            <speed>2400</speed>
        </HostAssetProcessor>
    </list>
</processor>
<volume>
    <list>
        <HostAssetVolume>
            <name>/dev</name>
            <size>328466432</size>
            <free>328466432</free>
        </HostAssetVolume>
    </list>
</volume>
<account>
    <list>
        <HostAssetAccount>
            <username>root</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>lxagent</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>omsagent</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>nonroot</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>giriraj</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>giri</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>nxautomation</username>
        </HostAssetAccount>
    </list>
</account>
<networkInterface>
    <list>
        <HostAssetInterface>

```

```
<hostname>ubuntu162.f004wtlrrhz2u3lcrb2sb010z2h.tx.internal.cloudapp.net</hostname>

    <interfaceName>eth0      Link encap</interfaceName>
    <macAddress>00:0d:3a:f2:4d:d9</macAddress>
    <type>LOCAL</type>
    <address>10.19.41.5</address>
    <gatewayAddress>10.19.41.1</gatewayAddress>
  </HostAssetInterface>
  <HostAssetInterface>
    <interfaceName>eth0      Link encap</interfaceName>
    <macAddress>00:0d:3a:f2:4d:d9</macAddress>
    <type>LOCAL</type>
    <address>fe80:0:0:0:20d:3aff:fef2:4dd9</address>
    <gatewayAddress>10.19.41.1</gatewayAddress>
  </HostAssetInterface>
  <HostAssetInterface>
    <interfaceName>eth1      Link encap</interfaceName>
    <macAddress>00:0d:3a:f2:57:25</macAddress>
    <type>LOCAL</type>
    <address>fe80:0:0:0:20d:3aff:fef2:5725</address>
    <gatewayAddress>10.19.41.1</gatewayAddress>
  </HostAssetInterface>
  <HostAssetInterface>
    <interfaceName>eth1      Link encap</interfaceName>
    <macAddress>00:0d:3a:f2:57:25</macAddress>
    <type>LOCAL</type>
    <address>10.19.41.13</address>
    <gatewayAddress>10.19.41.1</gatewayAddress>
  </HostAssetInterface>
</list>
</networkInterface>
</HostAsset>
</data>
</ServiceResponse>
```

### Example - HostAsset API output with GCP metadata info along with tags

API request:

```
curl -u "USERNAME:PASSWORD" -X GET
https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostasset/1841620
```

XML output:

```
<?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>1</count>
<data>
  <HostAsset>
    <id>1841620</id>
    <name>test-gcp-auto-1</name>
    <created>2018-04-23T18:11:05Z</created>
    <modified>2018-04-23T18:11:05Z</modified>
    <type>HOST</type>
    <tags>
      <list>
        <TagSimple>
          <id>7511024</id>
          <name>Cloud Agent</name>
        </TagSimple>
      </list>
    </tags>
    <sourceInfo>
      <list>
        <AssetSource/>
        <GcpAssetSourceSimple>
          <assetId>1841620</assetId>
          <type>GCP</type>
          <firstDiscovered>2018-04-
23T18:11:08Z</firstDiscovered>
          <lastUpdated>2018-04-23T18:11:08Z</lastUpdated>
          <gcpInstanceTags>
            <tags>
              <list>
                <GcpTags>
                  <key>name</key>
                  <value>test-tag</value>
                </GcpTags>
              </list>
            </tags>
          </gcpInstanceTags>
          <instanceId>GCP-1234567</instanceId>
          <hostname>test-gcp</hostname>
          <machineType>test</machineType>
          <zone>useast1</zone>
          <projectIdNo>12334</projectIdNo>
          <state>RUNNING</state>
          <projectId>5677</projectId>
          <network>testNetwork</network>
          <publicIpAddress>34.21.36.11</publicIpAddress>
          <privateIpAddress>10.10.10.10</privateIpAddress>
        </GcpAssetSourceSimple>
      </list>
    </sourceInfo>
    <dnsHostName>test-gcp-auto-1</dnsHostName>

```

```

<agentInfo>
  <agentId>12d31074-5c67-786a-b321-a8c34159daa9</agentId>
  <status>STATUS_INACTIVE</status>
  <lastCheckedIn>2018-04-23T11:05:04Z</lastCheckedIn>
  <connectedFrom>192.168.1.188</connectedFrom>
  <chirpStatus>Provisioned</chirpStatus>
  <platform>Linux</platform>
  <activatedModule>NONE</activatedModule>
  <agentConfiguration>
    <name>Dev Profile</name>
    <id>1001</id>
  </agentConfiguration>
  <activationKey>
    <activationId>d4f3c77b-55cd-43ea-8c08-
b9225cccb787</activationId>
    <title>agent-testing</title>
  </activationKey>
</agentInfo>
<netbiosName>test-gcp-auto-1</netbiosName>
<trackingMethod>QAGENT</trackingMethod>
</HostAsset>
</data>
</ServiceResponse>

```

## Schema updates

[https://<qualys-api-server>/qps/xsd/2.0/am/asset\\_source.xsd](https://<qualys-api-server>/qps/xsd/2.0/am/asset_source.xsd)

These new complexType names have been added: AzureAssetSourceSimple and GcpAssetSourceSimple

```

...
<complexType name="AzureAssetSourceSimple">
  <complexContent>
    <extension base="tns:AssetSource">
      <sequence>
        <element name="azureVmTags" type="tns:AzureVmTags" minOccurs="0" />
        <element name="name" type="string" minOccurs="0" />
        <element name="location" type="string" minOccurs="0" />
        <element name="vmSize" type="string" minOccurs="0" />
        <element name="vmId" type="string" minOccurs="0" />
        <element name="offer" type="string" minOccurs="0" />

        <element name="state" type="tns:AssetSourceStateCode" minOccurs="0" />
        <element name="publisher" type="string" minOccurs="0" />
        <element name="version" type="string" minOccurs="0" />
        <element name="osType" type="string" minOccurs="0" />
        <element name="subnet" type="string" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        <element name="ipv6" type="string" minOccurs="0" />
        <element name="subscriptionId" type="string" minOccurs="0" />
        <element name="resourceGroupName" type="string" minOccurs="0" />

        <element name="publicIpAddress" type="string" minOccurs="0" />
        <element name="privateIpAddress" type="string" minOccurs="0" />

    </sequence>
</extension>
</complexContent>
</complexType>

<complexType name="GcpAssetSourceSimple">
    <complexContent>
        <extension base="tns:AssetSource">
            <sequence>
                <element name="gcpInstanceTags" type="tns:GcpInstanceTags" minOccurs="0"
/>
                <element name="instanceId" type="string" minOccurs="0" />
                <element name="hostname" type="string" minOccurs="0" />
                <element name="machineType" type="string" minOccurs="0" />
                <element name="zone" type="string" minOccurs="0" />
                <element name="projectIdNo" type="string" minOccurs="0" />

                <element name="state" type="tns:AssetSourceStateCode" minOccurs="0" />
                <element name="projectId" type="string" minOccurs="0" />
                <element name="network" type="string" minOccurs="0" />
                <element name="macAddress" type="string" minOccurs="0" />

                <element name="publicIpAddress" type="string" minOccurs="0" />
                <element name="privateIpAddress" type="string" minOccurs="0" />

            </sequence>
        </extension>
    </complexContent>
</complexType>
...

```