Table of Contents

CloudView APIs ............................................................................................................................. 3

Getting Started with CloudView APIs ....................................................................................... 3

AWS APIs ..................................................................................................................................... 5

Assessment Reports .................................................................................................................... 39

Azure APIs ................................................................................................................................ 49

Connector Groups Management APIs ................................................................................... 76

Control Meta Data ..................................................................................................................... 84

GCP APIs ..................................................................................................................................... 92

Remediation .................................................................................................................................. 116

Reports ....................................................................................................................................... 119

Resource Inventory .................................................................................................................... 139

Alerting Response APIs ....................................................................................................... 160

User Access Management APIs .......................................................................................... 234

Secure IaC .................................................................................................................................... 241

Secure Infrastructure as Code ................................................................................................. 241

Pre-requisites & Template Support .......................................................................................... 243

Scanning Template Files Using CLI ....................................................................................... 244

Scanning Template Files Using APIs .................................................................................... 245

Understanding IaC Scan Output ............................................................................................. 257
CloudView APIs

Getting Started with CloudView APIs

Many CloudView features are available through REST APIs. You can use Swagger tool to access the REST APIs we support.

Accessing APIs Using Swagger

Swagger is a widely-adopted specification that allows for programmatically describing REST APIs. The Swagger UI provides all the details about the APIs and how to invoke them. This includes information like the HTTP verbs to use (GET, POST, PUT, etc.), the URL paths, allowable parameters and types, and so on.

You can directly access the Swagger UI from the following URL:

http://<QualysURL>/cloudview-api/swagger-ui.html

For example, if your account is on US Platform 2


Qualys Platforms

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

Qualys Platform URLs

Qualys US Platform 1 - https://qualysguard.qualys.com
Qualys US Platform 2 - https://qualysguard.qg2.apps.qualys.com
Qualys US Platform 3 - https://qualysguard.qg3.apps.qualys.com
Qualys EU Platform 1 - https://qualysguard.qualys.eu
Qualys EU Platform 2 - https://qualysguard.qg2.apps.qualys.eu
Qualys India Platform 1 - https://qualysguard.qg1.apps.qualys.in
Qualys Canada Platform - https://qualysguard.qg1.apps.qualys.ca

Do I need to Authenticate?

Authentication to the Qualys Cloud Platform is necessary before you try out the APIs.

Simply, click Authorize and provide the user name and password. You can now use the APIs!
AWS APIs

AWS Connector

We support the following operations for AWS Connector.

- Get list of connectors
- Get the details of a connector
- Get the AWS base account id
- Get the AWS Cloud Formation template
- Get the list of errors
- Create a new connector
- Run the provided connector
- Update the existing connector
- Enable Connector (AWS)
- Disable Connector (AWS)
- Delete the provided connectors

AWS Evaluations

We support the following control evaluations for AWS resources:

- Get the stats for specified control id and resource id
- Get the list of evaluations as per the account for AWS Controls
- Get the resources evaluated for the specified aws account and control id
Get list of AWS connectors

/rest/v1/aws/connectors

[GET]

List all AWS connectors in the user’s account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| filter    | Filter the connectors list by providing a query using Qualys syntax. The following search tokens are supported.  
- name: Name of the connector  
- description: Short description of the connector  
- state: Connector status The valid values are SUCCESS, PENDING, REGIONS_DISCOVERED, ERROR  
- connector.uuid: Unique Id assigned to the connector. For example, 6192ce15-e790-3fe2-a02c-b4bc75ecf123"  
- lastSyncedOn: Date and time when the connector synced with the cloud provider.  
Note: This time should be in UTC time |
| pageNo    | (integer) The page to be returned. |
| pageSize  | (integer) The number of records per page to be included in the response. |
| sort      | (keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc. |
Sample - Get list of AWS connectors in user's account

Return the list of all AWS connectors in the user’s scope.

**API request**

```
curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors?pageNo=0&pageSize=50'
```

**Response**

```
{
    "content": [
    {
        "name": "AWS Connector 2",
        "connectorId": "a7ad52b1-fb46-3baa-931f-4223a12a2ea7",
        "description": "",
        "provider": "AWS",
        "state": "SUCCESS",
        "totalAssets": 333,
        "lastSyncedOn": "Thu May 20 11:52:00 UTC 2021",
        "nextSyncedOn": "Thu May 20 13:50:52 UTC 2021",
        "remediationEnabled": true,
        "isGovCloud": false,
        "isChinaRegion": false,
        "awsAccountId": "XXXXXXXXXXXX",
        "accountAlias": "sample_account_alias",
        "isDisabled": false,
        "groups": [],
        "pollingFrequency": {
            "hours": 4,
            "minutes": 0
        },
        "error": "",
        "baseAccountId": "XXXXXXXXXXXX",
        "externalId": "USPOD01-4765-9011278609223",
        "arn": "arn:aws:iam::XXXXXXXXXXXXX:role/user_john_new_connector",
        "portalConnectorUuid": "2d39470f-cf33-45e3-8b12-ee5916bf18c9",
        "isPortalConnector": true
    }
    ],
    "pageable": {
    "sort": {
        "sorted": false,
        "unsorted": true
    }
    }
}```
Sample - Filter the list of AWS connectors in success state and sort in descending order with lastSyncedOn

API request

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors?filter=state%3DSUCCESS&pageNo=1&pageSize=50&sort=lastSyncedOn%3Adesc'

Response

{  "content": [    {      "name": "test",      "connectorId": "6192ce15-e790-3fe2-a02c-b4bc75ec1234",      "description": "sample description",      "provider": "AWS",      "state": "SUCCESS",      "totalAssets": 5484,      "lastSyncedOn": "Thu Nov 26 07:21:36 UTC 2020",      "nextSyncedOn": "Thu Nov 26 09:00:41 UTC 2020",      "remediationEnabled": true,      "isGovCloud": false,    }  ]}
"isDisabled": false,
"isChinaRegion": false,
"awsAccountId": "XXXXXXXXXXXX",
"account Alias": "alias-test",
"groups": [
  {
    "name": "group1",
    "uuid": "3ce54f33-81c6-30a2-b160-82e70cd1234"
  }
],
"pollingFrequency": {
  "hours": 7,
  "minutes": 0
},
"error": "",
"baseAccountId": "XXXXXXXXXXXX",
"externalId": "USPOD01-4765-9011278609223",
"arn": "arn:aws:iam::XXXXXXXXXXXX:role/test",
"isPortalConnector": false
},
...

"numberOfElements": 2,
"size": 50,
"number": 0
}
Get Connector Details

/rest/v1/aws/connectors/{connectorId}

[GET]

View details for a connector which is in the user’s scope.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user’s scope.</td>
</tr>
</tbody>
</table>

Sample - Get details of a specific connectors in user’s account

```
API request

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/a7ad52b1-fb46-3baa-931f-4223a12a2ea7'
```

Response

```json
{
    "name": "AWS Connector 2",
    "connectorId": "a7ad52b1-fb46-3baa-931f-4223a12a2ea7",
    "description": "",
    "provider": "AWS",
    "state": "SUCCESS",
    "totalAssets": 333,
    "lastSyncedOn": "Thu May 20 11:52:00 UTC 2021",
    "nextSyncedOn": "Thu May 20 13:50:52 UTC 2021",
    "remediationEnabled": true,
    "isGovCloud": false,
    "isChinaRegion": false,
    "awsAccountId": "XXXXXXXXXXXX",
    "accountAlias": "SAMPLE-ACCOUNT",
    "isDisabled": false,
    "groups": [
```
{
    "name": "AWS Connector",
    "uuid": "0ee655f8-2680-3ef1-98f7-d3d7b34aa485"
}

"pollingFrequency": {
    "hours": 2,
    "minutes": 0
},

"baseAccountId": "XXXXXXXXXXXX",
"externalId": "pod04-2722092-1626634575829",
"arn": "arn:aws:iam::XXXXXXXXXXXX:role/sample_role",
"isPortalConnector": false
Get AWS Base Account ID

/rest/v1/aws/connectors/awsBaseAccountId

[GET]

Fetches the AWS base account ID for you.

Sample - Get AWS Base Account ID

Fetches the AWS account ID. If there is a base account associated with your connector, the base account ID is reflected in response. Else, the Qualys account ID is displayed.

API request

curl -k -X GET -u <username>:<password> 
'https://<QualysURL>/rest/v1/aws/connectors/awsBaseAccountId'

Response

{
  "globalAccountId": "XXXXXXXXXXXXX",
  "chinaAccountId": "XXXXXXXXXXXXXXX",
  "govAccountId": "XXXXXXXXXXXXXXXXXXXX",
  "customerGlobalAccount": "false",
  "customerChinaAccount": "false",
  "customerGovAccount": "false"
}
Get Cloud Formation Template

/rest/v1/aws/connectors/aws/download

[GET]

Specify the External Id to be used for generating the AWS cloud formation template and download the template.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>awsCloudType</td>
<td>(string) AWS Cloud type used for generating the template for particular cloud.</td>
</tr>
<tr>
<td>externalId</td>
<td>(integer) External Id to be used for generating the template.</td>
</tr>
</tbody>
</table>

Sample - Download the Cloud Formation Template

**API request**

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/aws/download?awsCloudType=Gov&externalId=1532740312198'

**Response**
The Response includes a link to download the Cloud Formation Template.
Get Error List

/rest/v1/aws/connectors/{connectorId}/errors

[GET]

Get the list of errors encountered when executing connector.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user's scope.</td>
</tr>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Sample -

API request

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/226947d0-569d-11e9-8032-2fa7ed9d9b64/errors?pageNo=0&pageSize=50'

Response

```
{
  "content": [
    {
      "connectorName": "testts",
      "error": "Error getting config from af-south-1. Please check if region is enabled or config is enabled for this region",
      "occuredOn": "2020-07-09T12:41:50+0000",
      "region": null,
      "connectorId": "1e7fbcc0-89e7-11ea-a2c4-c9200d66f0b0"
    }
  ]
}```
{  "connectorName": "testts",
   "error": "Error getting config from eu-south-1. Please check if region is enabled or config is enabled for this region",
   "occuredOn": "2020-07-09T12:41:47+0000",
   "region": null,
   "connectorId": "1e7fbcc0-89e7-11ea-a2c4-c9200d66f0b0"
 }

"pageable": {
   "sort": {
      "sorted": false,
      "unsorted": true
   },
   "pageSize": 50,
   "pageNumber": 0,
   "offset": 0,
   "paged": true,
   "unpaged": false
 },
"last": true,
"totalElements": 2,
"totalPages": 1,
"first": true,
"sort": {
   "sorted": false,
   "unsorted": true
 },
"numberOfElements": 2,
"size": 50,
"number": 0
}
Create Connector (AWS)

/rest/v1/aws/connectors

[POST]

Specify the connector details such as qualysAccountId, arn, externalId, and so on and create a new connector.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorBody</td>
<td>(body) Specify the connector details such as qualysAccountId, arn, externalId, and so on. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```json
{
    "arn": "string",
    "description": "string",
    "externalId": "string",
    "isChinaRegion": true,
    "isGovCloud": false,
    "isPortalConnector": true,
    "name": "string",
    "remediationEnabled": true,
    "pollingFrequency": {
        "hours": 0,
        "minutes": 0
    }
}
```

Where,

- `arn`: Specify the ARN of the cross-account role you created in your AWS account.

- `description` is optional and you can give a short description stating the purpose of the connector you
want to create.

-externalId: Specify the unique external ID. The valid format for the external ID is `<Qualys POD>-<Qualys Subscription ID>-<configurable alphanumeric number>`

where,

Qualys POD refers to the Qualys Platform associated with your Qualys subscription. To know the Qualys Platform you belong to, refer to [https://www.qualys.com/platform-identification/](https://www.qualys.com/platform-identification/).

Qualys Subscription ID: Your unique Qualys Subscription ID.

configurable alphanumeric number: Unique random alphanumeric number You can use a combination of alphabets (a-z, A-Z) and numbers to generate the unique number. You can use minimum 5 or maximum 13 digits to complete the external ID combination in the new format.

Note: Special characters are not permitted in the random number.

-isChinaRegion (boolean): A flag indicating whether the Connector also is created China region or not. Set this flag to true to create the connector for China.

-isGovCloud (boolean): A flag indicating whether the Connector also is created GovCloud region or not. Set this flag to true to create the connector for GovCloud.

Note: You can set either isChinaRegion or isGovCloud to true for one connector. If both are set to false, the connector is created for Global region.

-isPortalConnector: (boolean). A flag indicating whether the Connector also is created in Portal module or not (Asset View). If the connector is created in AssetView as well, then the authentication information associated with the connector is linked to CloudView as well. If you update the authentication information for the connector
in AssetView, it will automatically reflect in CloudView as well.

-name is the name for the connector you want to create.

-remediationEnabled (boolean). A flag to enable or disable remediation for the connector. To enable remediation, the remediationEnabled flag needs to be set to true. This flag is optional. By default, this flag is configured to false. If remediation is enabled on a connector, the response includes "remediationEnabled": true. If you do not set the flag, the existing remediation status of the connector is displayed in the response.

Note: The remediation feature is available only to Cloud Security Assessment (CSA) subscribers and is enabled by default.

- pollingFrequency: Polling frequency for a connector decides the rate at which the connector should poll the cloud provider and fetch the data.

- hours: Specify the time in hours. The valid range is 1 to 24 hours.

- minutes: Specify the time in minutes. The valid range is 0 to 59 minutes.

You can configure frequency from minimum one hour to maximum 24 hours. We recommend that you configure frequency of 4 hours or more for optimal use of your connector. Configuring a low polling frequency (lesser than 4 hours) can affect the performance of the connector and may result in AWS API throttling error.

Note: Configuration of connector polling frequency is enabled only for Cloud Security Assessment (CSA) users. For all other users, the default connector polling frequency is pre-configured and even if you update the connector polling frequency, the change in frequency will not reflect. For Cloud Inventory (CI) users, the pre-configured frequency is 24 hours. For the trial period, the pre-configured frequency it is 4 hours.
Sample - Create a connector in CloudView

Create a connector in the user’s scope. A copy of the same connector is also created in AssetView provided we set "isPortalConnector": true. If the connector already exists in AssetView, then set “isPortalConnector”: false.

**API request**

```
curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors'
```

**Request POST Data**

```
{
    "arn": "arn:aws:iam::XXXXXXXXXXXX:role/CloudViewTest",
    "description": "Sample External ID",
    "externalId": "USPOD01-4765-9011278609223",
    "isChinaRegion": true,
    "isGovCloud": true,
    "isPortalConnector": true,
    "name": "ABC",
    "remediationEnabled": true,
    "pollingFrequency": {
        "hours": 1,
        "minutes": 1
    }
}
```

**Response**

```
{
    "name": "ABC",
    "connectorId": "8889b18b-6f92-33d3-8afd-6d9d144177e8",
    "description": "Sample External ID",
    "provider": "AWS",
    "state": "PENDING",
    "totalAssets": 0,
    "lastSyncedOn": "Wed Jan 13 07:24:46 UTC 2021",
    "nextSyncedOn": "Wed Jan 13 08:25:46 UTC 2021",
    "isGovCloud": true,
    "isChinaRegion": true,
    "awsAccountId": "XXXXXXXXXXXX",
    "isDisabled": false,
    "groups": [],
    "pollingFrequency": {
        "hours": 1,
```
"minutes": 1,
"error": "50004",
"remediationEnabled": true,
"baseAccountId": "XXXXXXXXXXXX",
"externalId": "USPOD01-4765-9011278609223",
"arn": "arn:aws:iam::XXXXXXXXXXXX:role/CloudViewTest",
"isPortalConnector": false
Run Connector

/rest/v1/aws/connectors/run

[POST]

Specify the IDs of the connectors that you want to run.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorRunRequest</td>
<td>(body) Specify the IDs of the connectors that you want to execute/run.</td>
</tr>
</tbody>
</table>

Example:

```
[  "string"
]
```

Sample - Get details of a specific connectors in user’s account

**API request**

curl -k -X POST -u <username>:<password> -d '"
["5f83c570-51e6-11e9-bd82-c173b8d28354"]"
' 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/run'

**Response**

No Content
Response Code: 204
Update Connector (AWS)

/rest/v1/aws/connectors/{connectorId}

[PUT]

Specify the connector ID and you can then update details of the specified connector.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user’s scope.</td>
</tr>
<tr>
<td>connectorBody</td>
<td>(body) Specify the connector details such as qualysAccountId, arn, externalId, and so on. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```json
{
    "arn": "string",
    "description": "string",
    "externalId": "string",
    "isChinaRegion": true,
    "isGovCloud": false,
    "isPortalConnector": true,
    "name": "string",
    "remediationEnabled": true
    "pollingFrequency": {
        "hours": 0,
        "minutes": 0
    }
}
```

Where,

- arn: Specify the ARN of the cross-account role you created in your AWS account.
description is optional and you can give a short description stating the purpose of the connector you want to create.

-externalId: Specify the unique external ID. The valid format for the external ID is <Qualys POD>-<Qualys Subscription ID>-<configurable alphanumeric number> where,

Qualys POD refers to the Qualys Platform associated with your Qualys subscription. To know the Qualys Platform you belong to, refer to [https://www.qualys.com/platform-identification/](https://www.qualys.com/platform-identification/).

Qualys Subscription ID: Your unique Qualys Subscription ID.

configurable alphanumeric number: Unique random alphanumeric number You can use a combination of alphabets (a-z, A-Z) and numbers to generate the unique number. You can use minimum 5 or maximum 13 digits to complete the external ID combination in the new format.

Note: Special characters are not permitted in the random number.

-isChinaRegion (boolean): A flag indicating whether the Connector also is created China region or not. Set this flag to true to create the connector for China.

-isGovCloud (boolean): A flag indicating whether the Connector also is created GovCloud region or not. Set this flag to true to create the connector for GovCloud.

**Note:** You can set either isChinaRegion or isGovCloud to true for one connector. If both are set to false, the connector is created for Global region.

-isPortalConnector: (boolean). A flag indicating whether the Connector also is created in Portal module or not.
(Asset View). If the connector is created in AssetView as well, then the authentication information associated with the connector is linked to CloudView as well. If you update the authentication information for the connector in AssetView, it will automatically reflect in CloudView as well.

-name is the name for the connector you want to create.

-remediationEnabled (boolean). A flag to enable or disable remediation for the connector. To enable remediation, the remediationEnabled flag needs to be set to true. This flag is optional. By default, this flag is configured to false. If remediation is enabled on a connector, the response includes "remediationEnabled": true. If you do not set the flag, the existing remediation status of the connector is displayed in the response.

Note: The remediation feature is available only to Cloud Security Assessment (CSA) subscribers and is enabled by default.

- pollingFrequency: Polling frequency for a connector decides the rate at which the connector should poll the cloud provider and fetch the data.

- hours: Specify the time in hours. The valid range is 1 to 24 hours.

- minutes: Specify the time in minutes. The valid range is 0 to 59 minutes.

You can configure frequency from minimum one hour to maximum 24 hours. We recommend that you configure frequency of 4 hours or more for optimal use of your connector. Configuring a low polling frequency (lesser than 4 hours) can affect the performance of the connector and may result in AWS API throttling error.

Sample - Update AWS Connector

Let us consider an example to update the description of the connector.
API request

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/8889b18b-6f92-33d3-8afd-6d9d144177e8'

Request POST Data

```json
{
    "arn": "arn:aws:iam::XXXXXXXXXXXX:role/CloudViewTest",
    "description": "Update External ID",
    "externalId": "USP0D01-4765-9011278609223",
    "isChinaRegion": true,
    "isGovCloud": true,
    "isPortalConnector": true,
    "name": "ABC",
    "remediationEnabled": true,
    "pollingFrequency": {
        "hours": 1,
        "minutes": 1
    }
}
```

Response

```json
{
    "name": "My updated AWS Connector",
    "connectorId": "8889b18b-6f92-33d3-8afd-6d9d144177e8",
    "description": "Update External ID",
    "provider": "AWS",
    "state": "PENDING",
    "totalAssets": 0,
    "lastSyncedOn": "Wed Jan 13 07:28:39 UTC 2021",
    "nextSyncedOn": "Wed Jan 13 08:29:40 UTC 2021",
    "isGovCloud": true,
    "isChinaRegion": true,
    "awsAccountId": "XXXXXXXXXXXX",
    "isDisabled": false,
    "groups": [],
    "pollingFrequency": {
        "hours": 1,
        "minutes": 1
    },
    "error": "50004",
    "remediationEnabled": true,
}
```
"baseAccountId": "XXXXXXXXXXXX",
"externalId": "USPOD01-4765-9011278609223",
"arn": "arn:aws:iam::XXXXXXXXXXXX:role/CloudViewTest",
"isPortalConnector": false
}
Enable Connector (AWS)

/rest/v1/aws/connectors/connectors/enable

[PATCH]

We give you the flexibility to enable or disable a connector. By default, all connectors you create are in enabled state. When you disable a connector, it is not eligible for auto-run or manual run. You can enable a disabled connector to make it eligible for auto-run or manual run. You can view connector details, update or delete a disabled connector.

Note: If a combination of connectors (enabled/disabled) or only disabled connector are run using Run API request call, the response is always acknowledged: true. The Run connector API always skips disabled connector for all clouds.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user's scope to be enabled.</td>
</tr>
</tbody>
</table>

Sample - Enable AWS Connector

Let us consider an example to enable an AWS connector.

**API request**

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/connectors/enable' -d "[ "359dba68-8559-30f4-88d5-d4720647a23f"]"

**Response**

```json
{
  "acknowledged": true
}
```
Disable Connector (AWS)

/rest/v1/aws/connectors/connectors/disable

[PATCH]

We give you the flexibility to enable or disable a connector. By default, all connectors you create are in enabled state. When you disable a connector, it is not eligible for auto-run or manual run. You can enable a disabled connector to make it eligible for auto-run or manual run. You can view connector details, update or delete a disabled connector.

Note: If a combination of connectors (enabled/disabled) or only disabled connector are run using Run API request call, the response is always acknowledged: true. The Run connector API always skips disabled connector for all clouds.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user's scope to be disabled.</td>
</tr>
</tbody>
</table>

Sample - Disable AWS Connector

Let us consider an example to disable an AWS connector.

**API request**

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/connectors/disable'-d "[ "359dba68-8559-30f4-88d5-d4720647a23f"]"

**Response**

{  "acknowledged": true}
CloudView APIs

}
Delete Connector (AWS)

/rest/v1/aws/connectors

[DELETE]

Delete the specified connector which is in the user’s scope.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user's scope.</td>
</tr>
</tbody>
</table>

Sample - Delete AWS connector in user's account

API request

curl -X DELETE -u <username>:<password>
-d '"215947d0-569d-11e9-8032-2fa7ed9d9b64"
' https://<QualysURL>/cloudview-api/rest/v1/aws/connectors'

Response

No Content
Response Code: 204
AWS Evaluations

Get the stats for specified control id and resource id

/rest/v1/aws/evaluations/stats/{controlId}/{connectorId}?resourceId={resourceId}

[GET]

Specify the details such as control ID, resource ID, and connector ID to get the statistics for specified control and resource ID.

Note: By default, the response includes the data for last 24 hours.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>controlId</td>
<td>(string) Specify the control ID of a control for which resources evaluated need to be fetched.</td>
</tr>
<tr>
<td>resourceId</td>
<td>(string) Specify the unique ID of the resource being evaluated.</td>
</tr>
<tr>
<td>connectorId</td>
<td>(string) Specify the unique Id associated with the connector in the user’s scope.</td>
</tr>
</tbody>
</table>

Sample - Get the statistics for a specified control and resource

API request


Response

{  "firstEvaluated": "2021-03-08T23:01:10+0000",  "lastEvaluated": "2021-03-22T12:07:05+0000",}
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dateReopen</td>
<td>&quot;2021-03-25T08:57:17+0000&quot;</td>
</tr>
<tr>
<td>dateFixed</td>
<td>&quot;2021-03-25T08:51:45+0000&quot;</td>
</tr>
</tbody>
</table>
Get the list of evaluations as per the account for AWS Controls

/rest/v1/aws/evaluations/{accountId}

[GET]

Specify the details such as control ID, resource ID, and connector ID to get the statistics for specified control and resource ID.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accountId</td>
<td>(string) Specify the unique Id associated with your AWS account.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter the resources list by providing a query using Qualys syntax. If you do not add a date filter, by default the data for last 24 hours is included in the response. If you need data for specific date or date range, form your filter query using evaluatedOn token. Examples:</td>
</tr>
<tr>
<td></td>
<td>Show resources discovered within certain dates evaluatedOn: [2019-01-01 ... 2019-03-01]</td>
</tr>
<tr>
<td></td>
<td>Show resources updated starting 2019-01-01, ending 1 month ago evaluatedOn: [2019-01-01 ... now-1m]</td>
</tr>
<tr>
<td></td>
<td>Show resources updated starting 2 weeks ago, ending 1 second ago evaluatedOn: [now-2w ... now-1s]</td>
</tr>
<tr>
<td></td>
<td>Show resources discovered on specific date evaluatedOn: 2019-01-08</td>
</tr>
</tbody>
</table>

**Sample - Get the list of evaluations as per the account for AWS Controls**
CloudView APIs

**API request**

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/evaluations/888888888888'

**Response**

```json
{
    "content": [
    {
        "controlName": "Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password",
        "policyName": "CIS Amazon Web Services Foundations Benchmark",
        "criticality": "HIGH",
        "service": "IAM",
        "result": "FAIL",
        "controlId": "1",
        "passedResources": 6,
        "failedResources": 32,
        "passWithExceptionResources": 1
    },
    {
        "controlName": "Ensure console credentials unused for 90 days or greater are disabled",
        "policyName": "CIS Amazon Web Services Foundations Benchmark",
        "criticality": "HIGH",
        "service": "IAM",
        "result": "FAIL",
        "controlId": "2",
        "passedResources": 8,
        "failedResources": 30,
        "passWithExceptionResources": 0
    },
    ...
    {
        "controlName": "Ensure that all the expired SSL/TLS certificates stored in AWS IAM are removed",
        "policyName": "AWS Best Practices Policy",
        "criticality": "HIGH",
        "service": "IAM",
        "result": "FAIL",
        "controlId": "68",
        "passedResources": 1,
        "failedResources": 3
    }
}
"passWithExceptionResources": 0
]
,"pageable": {
 "sort": {
   "unsorted": true,
   "sorted": false
 },
 "pageSize": 100,
 "pageNumber": 0,
 "offset": 0,
 "paged": true,
 "unpaged": false
 },
 "last": true,
 "totalElements": 68,
 "totalPages": 1,
 "first": true,
 "sort": {
   "unsorted": true,
   "sorted": false
 },
 "numberOfElements": 68,
 "size": 100,
 "number": 0
}
Get the resources evaluated for the specified AWS account and control ID:

/rest/v1/aws/evaluations/{accountId}/resources/{controlId}

[GET]

Specify the details such as account ID, resource ID, and connector ID to get the statistics for specified control and resource ID.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accountId</td>
<td>(string) Specify the unique ID associated with your AWS account.</td>
</tr>
<tr>
<td>controlId</td>
<td>(string) Specify the control ID of a control for which resources evaluated need to be fetched</td>
</tr>
</tbody>
</table>

**Filter**

Filter the resources list by providing a query using Qualys syntax.

If you do not add a date filter, by default the data for last 24 hours is included in the response. If you need data for specific date or date range, form your filter query using evaluatedOn token.

Examples:

Show resources discovered within certain dates
```
evaluatedOn: [2019-01-01 ... 2019-03-01]
```

Show resources updated starting 2019-01-01, ending 1 month ago
```
evaluatedOn: [2019-01-01 ... now-1m]
```

Show resources updated starting 2 weeks ago, ending 1 second ago
```
evaluatedOn: [now-2w ... now-1s]
```

Show resources discovered on specific date
<table>
<thead>
<tr>
<th><strong>evaluatedOn</strong>: 2019-01-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageNo (integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize (integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

**Sample - Get all or filter the evaluations for your account**

**API request**

```bash
```

**Response**

```json
{
  "content": [
    {
      "resourceId": "Default",
      "region": "us-east-1",
      "accountId": "111111111111",
      "evaluatedOn": "2020-12-15T06:42:18+0000",
      "evidences": [
        {
          "settingName": "KMS Key ID",
          "actualValue": null
        }
      ],
      "resourceType": "CLOUD_TRAIL",
      "connectorId": "6192ce15-e790-3fe2-a02c-b4bc75ecf199",
      "result": "PASS_WITH_EXCEPTION",
      "evaluationDates": {
        "firstEvaluated": "2020-12-08T02:27:24+0000",
        "lastEvaluated": "2020-12-15T06:42:18+0000",
        "dateReopen": null,
        "dateFixed": "2020-12-10T13:58:47+0000"
      }
    }
  ]
}
```

...
"last": true,
"totalPages": 1,
"totalElements": 2,
"first": true,
"sort": {
  "sorted": false,
  "unsorted": true
},
"numberOfElements": 2,
"size": 50,
"number": 0
Assessment Reports

Create Assessment Report

/rest/v1/report/assessment/create

[POST]

Specify the report settings such as policy details, connector details, cloud provider and so on to generate assessment reports. You can generate the assessment reports to view the compliance posture of the organization.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createReportRequest</td>
<td>(body) You need to provide the details required to generate the report in the createReportRequest parameter. The syntax for the same is given below:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
</tbody>
</table>
|                            |   "reportName": "string",
|                            |   "description": "string",
|                            |   "format": "string",
|                            |   "resourceSummaryInclude": true,
|                            |   "cloudType": "string",
|                            |   "query": "string",
|                            |   "startDate": "string",
|                            |   "endDate": "string",
|                            |   "executionType": "RUN_TIME",
|                            |   "policyIds": [                                                                                                                                   |
|                            |     "string"
|                            |   ],                                                                                                                                           |
|                            |   "resourceResults": [                                                                                                                                |
|                            |     "PASS"
|                            |   ],                                                                                                                                              |
|                            |   "groupIds": [                                                                                                                                   |
|                            |     "string"
|                            |   ]                                                                                                                                              |
"connectorIds": ["string" ]
}

where,

**reportName**: name of the report

**description**: short description stating the purpose of the report you want to create.

**format**: the report format: CSV or PDF.

**resourceSummaryInclude**: Set to true to include details resource ID, connector, control ID, resource type, evaluation date, and resource result in the report (applicable only for PDF report format).

Note: Assessment reports containing up to 8k records with Resource Summary get successfully downloaded. Assessment report exceeding 8k records and Resource Summary is currently not supported for PDF reports.

**cloudType**: the cloud provider (AWS, Azure, or GCP)

**query**: Form a search query for your report using Qualys Query Language (QQL). For more information on query formation, see the “How to Search” topic in the CloudView online help.

The **startDate** and **endDate** parameters are mandatory. Use them as date filter for your query. Specify the date in yyyy-mm-ddTHH:MM:SSZ format. For example, to use specify 14th October 2020 at 3.26 PM, you need to use 2020-10-14T03:26:15Z format.

**executionType**: Set to one of the following values to generate the report depending on the execution
type of the controls.

Set to RUN_TIME to generate assessment report with control evaluations for deployed cloud resources.

Set to BUILD_TIME to generate assessment report with control evaluations for cloud resources within the IaC templates.

**policyId**: unique ID associated with the policy.

For CSV report format, you can specify multiple policy IDs.

For PDF report format, you can specify only one policy ID.

**resourceResults**: the evaluation results to be included in the reports for resources evaluated against the controls that meet criteria defined in Search Query. You could specify Pass, PassE (pass with exceptions), and Fail options. You can specify multiple options.

**groupId**: unique Id of the (connector) group.

**connectorIds**: unique Id associated with the connector.

Sample - Create an assessment report

**API request**

curl -k -X POST -u <username>:<password>  
'https://<QualysURL>/cloudview-api/rest/v1/report/assessment/create

**Request POST Data**

```json
{
  "reportName": "Sample PDF Report",
  "description": "PDF report format",
  "format": "pdf",
  "resourceSummaryInclude": false,
```
"cloudType": "GCP",
"query": "",
"startDate": "2021-01-24T08:46:36Z",
"endDate": "2021-01-25T08:46:36Z",
"policyIds": ["636335f0-4730-11ea-8758-77aa7bc96f55"],
"resourceResults": ["PASS"],
"groupIds": [],
"connectorIds": []
}

Response
1252bf70-0ee3-11eb-8be0-19cb59be89b6
The response returns the unique report ID on successfully creating the report.

Sample - Create an assessment report with Run Time controls

API request

curl -X POST -u <username>:<password>
'https://<QualysURL>/cloudview-api/rest/v1/report/assessment/create'

Request POST Data
{
  "cloudType": "AWS",
  "connectorIds": ["XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"],
  "description": "Sample Assessment Report",
  "endDate": "2021-12-30T04:49:22Z",
  "executionType": "RUN_TIME",
  "format": "CSV",
  "groupIds": ["3fa85f64-5717-4562-b3fc-2c963f66afa6"],
  "iacResourceResults": ["FAIL"],
  "policyIds": ["3fa85f64-5717-4562-b3fc-2c963f66afa6"],
  "query": "",
  "reportName": "Sample Assessment Report CSV",
  "resourceResults": ["FAIL"],
  "resourceSummaryInclude": true,
  "startDate": "2021-12-27T04:49:22Z"
}

Response
1252bf70-0ee3-11eb-8be0-19cb59be89b6
The response returns the unique report ID on successfully creating the report.

Sample - Create an assessment report with Build Time controls

**API request**

curl -X POST -u <username>:<password> 
'https://<QualysURL>/cloudview-api/rest/v1/report/assessment/create'

**Request POST Data**

```
{
  "cloudType": "AWS",
  "connectorIds": [
    "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX"
  ],
  "description": "Sample Assessment Report",
  "endDate": "2021-12-30T04:49:22Z",
  "executionType": "BUILD_TIME",
  "format": "CSV",
  "groupIds": [
    "3fa85f64-5717-4562-b3fc-2c963f66afa6"
  ],
  "iacResourceResults": [
    "FAIL"
  ],
  "policyIds": [
    "3fa85f64-5717-4562-b3fc-2c963f66afa6"
  ],
  "query": "",
  "reportName": "Sample Assessment Report CSV",
  "resourceResults": [
    "FAIL"
  ],
  "resourceSummaryInclude": true,
  "startDate": "2021-12-27T04:49:22Z"
}
```

**Response**

3452bf70-0ee3-11eb-8be0-19cb59be89b6
The response returns the unique report ID on successfully creating the report.
Get List Assessment Report

/rest/v1/report/assessment/list

[GET]

You can now fetch the list of all the assessment reports in your account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createdBy</td>
<td>Use values to find reports created by a certain user.</td>
</tr>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>reportName</td>
<td>(string) Name of the report.</td>
</tr>
<tr>
<td>sortBy</td>
<td>Specify the field that decides the sort order for the rules.</td>
</tr>
<tr>
<td>reportId</td>
<td>(string) Specify the unique Id associated with the report.</td>
</tr>
<tr>
<td>status</td>
<td>(string) Specify the report status from Accepted, Completed, Failed, Generated, Processing.</td>
</tr>
</tbody>
</table>

Sample - Get list of assessment reports

API request


Response
{  
  "reportName": "Assessment Report in PDF",
  "status": "COMPLETED",
  "fileFormat": "pdf",
  "createdAt": "2021-01-25T06:04:27.000Z",
  "createdBy": "user_john",
  "templateName": "Assessment Report",
  "reportType": "On-Demand",
  "expiresOn": "2021-02-02",
  "reportId": "2e383cc0-5ed3-11eb-ac38-9b1adcf6e2ef"
}

Re-run Assessment Report

/rest/v1/report/assessment/{reportId}/rerun

[POST]

You can execute an existing assessment report using the unique report Id assigned to a report. The same report configurations are retained in the report you run again and a new report is generated with a new report Id.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportId</td>
<td>(string) Specify the unique Id associated with the report.</td>
</tr>
</tbody>
</table>

Sample - Re-run an assessment report

**API request**

```bash
```

**Response**

`1252bf70-0ee3-11eb-8be0-19cb59be89b6`

The response returns the a new unique report ID on successfully execution the report.
Download Assessment Report

/rest/v1/report/assessment/{reportId}/download

[GET]

You can download a specific report in CSV or PDF format using the report Id. You can download report for IaC posture to know the evaluation results and prevent misconfigurations.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportId</td>
<td>(string) Specify the unique Id associated with the report.</td>
</tr>
<tr>
<td>reportFormat</td>
<td>Specify the report format: csv or pdf.</td>
</tr>
</tbody>
</table>

Sample - Download assessment reports (CSV format)

**API request**


**Response**

Response Code: 200
The response includes the assessment report in CSV format.

Sample - Download assessment reports in PDF format

**API request**


**Response**
Response Code: 200
The response body includes the link for download of assessment report in PDF format.
Azure APIs

Azure Connector

We support the following operations for Azure Connector.

- Get list of connectors
- Get the details of a connector
- Create a new connector
- Run the provided connector
- Update the existing connector
- Enable Connector (Azure)
- Disable Connector (Azure)
- Delete the provided connectors

Azure Evaluations

We support the following control evaluations for Azure resources:

- Get the statistics for specified control and resource
- Get the list of evaluations as per account for Azure controls
- Get the resources evaluated for specified Azure account Id and control Id
Get Azure Connectors

/rest/v1/azure/connectors

[GET]

List all Azure connectors in the user’s account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| filter    | Filter the connectors list by providing a query using Qualys syntax. The following search tokens are supported.  
- name: Name of the connector  
- description: Short description of the connector  
- state: Connector status The valid values are SUCCESS, PENDING, REGIONS_DISCOVERED, ERROR  
- connector.uuid: Unique Id assigned to the connector. For example, 6192ce15-e790-3fe2-a02c-b4bc75ecf123"  
- lastSyncedOn: Date and time when the connector synced with the cloud provider.  
Note: This time should be in UTC time |
| pageNo    | (integer) The page to be returned. |
| pageSize  | (integer) The number of records per page to be included in the response. |
| sort      | (keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc. |
Sample - Get list of Azure connectors in user's account

Return the list of all connectors in the user’s scope.

### API request

```
curl -X GET -u <username>:<password>
'https://<QualysURL>/cloudviewapi/rest/v1/azure/connectors/2e0c1660-d061-11e9-ad71-df4fba75b3c5'
```

### Response

```
{
  "content": [
    {
      "name": "Azure Connector",
      "connectorId": "2725fd62-c0f7-3ba0-b714-2595fe19e734",
      "description": "",
      "provider": "AZURE",
      "state": "SUCCESS",
      "totalAssets": 213,
      "lastSyncedOn": "Thu May 20 11:51:51 UTC 2021",
      "nextSyncedOn": "Thu May 20 15:40:41 UTC 2021",
      "remediationEnabled": false,
      "isGovCloud": false,
      "isDisabled": false,
      "applicationId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "subscriptionName": "Sample_account",
      "groups": [
        {
          "name": "Azure Group",
          "uuid": "d24f8981-c3ad-37ae-8496-29ee5a459070"
        },
        {
          "name": "Sales Group",
          "uuid": "4659b745-c06d-39ef-8bc1-e6f7f5acee12"
        }
      ],
      "pollingFrequency": {
        "hours": 4,
        "minutes": 0
      },
      "directoryId": "ff4e2413-65ab-4dc2-9e5b-1ea02d3d94eb"
    }
  ]
}
```
"name": "Azure Conn2",
"connectorId": "6b0b7c71-d0f3-3226-a88a-b0157ea47c19",
"description": "",
"provider": "AZURE",
"state": "SUCCESS",
"totalAssets": 49,
"lastSyncedOn": "Thu May 20 11:51:15 UTC 2021",
"nextSyncedOn": "Thu May 20 14:01:15 UTC 2021",
"remediationEnabled": false,
"isGovCloud": false,
"isDisabled": false,
"applicationId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
"subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
"subscriptionName": "sample_account",
"groups": [],
"pollingFrequency": {
  "hours": 4,
  "minutes": 0
},
"nextSyncedOn": "Fri May 20 18:33:08 UTC 2021",
}]
"pageable": {
  "sort": {
  "sorted": true,
  "unsorted": false
},
  "pageSize": 50,
  "pageNumber": 0,
  "offset": 0,
  "paged": true,
  "unpaged": false
},
"last": true,
"totalPages": 1,
"totalElements": 2,
"first": true,
"sort": {
  "sorted": true,
  "unsorted": false
},
"numberOfElements": 2,
"size": 50,
"number": 0}
Sample - Filter the list of Azure connectors in success state and sort in descending order with lastSyncedOn

**API request**

curl -k -X GET -u <username>:<password> "https://<QualysURL>/cloudview-api/rest/v1/azure/connectors?filter=state%3DSUCCESS&pageNo=1&pageSize=50&sort=lastSyncedOn%3Adesc"

**Response**

```json
{
  "content": [
    {
      "name": "test",
      "connectorId": "d56884c8-de42-3d62-a5ea-5d46c2412345",
      "description": "azure connector",
      "provider": "AZURE",
      "state": "SUCCESS",
      "totalAssets": 117,
      "lastSyncedOn": "Thu Nov 26 07:55:26 UTC 2020",
      "nextSyncedOn": "Thu Nov 26 11:55:26 UTC 2020",
      "remediationEnabled": true,
      "isGovCloud": false,
      "applicationId": "d8c3a45a-e6f9-449e-8a54-2416e2d12345",
      "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "subscriptionName": "test-subscription-name",
      "groups": [],
      "pollingFrequency": {
        "hours": 4,
        "minutes": 0
      },
      "directoryId": "ff4e2413-65ab-4dc2-9e5b-1ea02d312345"
    }
  ]
}
```
Get Connector Details (Azure)

/rest/v1/azure/connectors/{connectorId}

[GET]

View details for a specific Azure connector which is in the user’s scope.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(string) Specify the unique Id associated with connector in the user’s scope.</td>
</tr>
</tbody>
</table>

Sample - Get details of a specific Azure connector in user's account

**API request**

curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQK==' 'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors/2725fd62-c0f7-3ba0-b714-2595fe19e734'

**Response**

```
{
    "name": "Azure Connector",
    "connectorId": "2725fd62-c0f7-3ba0-b714-2595fe19e734",
    "description": "",
    "provider": "AZURE",
    "state": "SUCCESS",
    "totalAssets": 213,
    "lastSyncedOn": "Thu May 20 11:51:51 UTC 2021",
    "nextSyncedOn": "Thu May 20 15:40:41 UTC 2021",
    "remediationEnabled": false,
    "isGovCloud": false,
    "isDisabled": false,
    "applicationId": "f076c321-694d-4929-ae0b-d2bd14d1a4d7",
    "subscriptionId": "9de9e0a7-4f67-4812-917d-2246853844e1",
```
"subscriptionName": "qlys-dev-cvdev",
"groups": [
  {
    "name": "QA Group",
    "uuid": "4659b745-c06d-39ef-8bc1-e6f7f5acee12"
  },
  {
    "name": "Azure Group",
    "uuid": "d24f8981-c3ad-37ae-8496-29ee5a459070"
  }
],
"pollingFrequency": {
  "hours": 4,
  "minutes": 0
},
"directoryId": "ff4e2413-65ab-4dc2-9e5b-1ea02d3d94eb"}
Create Connector (Azure)

/rest/v1/azure/connectors

[POST]

Specify the connector details such as application Id, authenticationKey, description, directoryId, name, and subscription Id of your Azure account and create a new connector.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorBody</td>
<td>(body) Specify the connector details such as qualysAccountId, arn, externalId, and so on. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```json
{
  "applicationId": "string",
  "authenticationKey": "string",
  "description": "string",
  "directoryId": "string",
  "isGovCloud": false,
  "name": "string",
  "remediationEnabled": true,
  "subscriptionId": "string",
  "pollingFrequency": {
    "hours": 0,
    "minutes": 0
  }
}
```

Where,

- applicationId: Unique identifier of the application you create on Azure portal.
- authenticationKey: The secret key generated after you provide permission to the application to access the
CloudView APIs

Windows Azure Service.

- description is optional and you can give a short description stating the purpose of the connector you want to create.

- directoryId: Unique identifier of your Azure Active Directory.

- isGovCloud (boolean): A flag indicating whether the Connector also is created GovCloud region or not. Set this flag to true to create the connector for GovCloud.

- name is the name for the connector you want to create.

- remediationEnabled (boolean). A flag to enable or disable remediation for the connector. To enable remediation, the remediationEnabled flag needs to be set to true. This flag is optional. By default, this flag is configured to false. If remediation is enabled on a connector, the response includes "remediationEnabled": true. If you do not set the flag, the existing remediation status of the connector is displayed in the response.

Note: The remediation feature is available only to Cloud Security Assessment (CSA) subscribers and is enabled by default.

- subscriptionId: Unique identifier of your Microsoft Azure subscription.

- pollingFrequency: Polling frequency for a connector decides the rate at which the connector should poll the cloud provider and fetch the data.

- hours: Specify the time in hours. The valid range is 1 to 24 hours.

- minutes: Specify the time in minutes. The valid range is 0 to 59 minutes.

You can configure frequency from minimum one hour
to maximum 24 hours. We recommend that you configure frequency of 4 hours or more for optimal use of your connector. Configuring a low polling frequency (lesser than 4 hours) can affect the performance of the connector and may result in Microsoft Azure API throttling error.

Note: Configuration of connector polling frequency is enabled only for Cloud Security Assessment (CSA) users. For all other users, the default connector polling frequency is pre-configured and even if you update the connector polling frequency, the change in frequency will not reflect. For Cloud Inventory (CI) users, the pre-configured frequency is 24 hours. For the trial period, the pre-configured frequency it is 4 hours.

Sample - Create a Azure Connector

**API request**

curl -k -X POST -u <username>:<password> 
'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors'

**Request POST Data**

```json
{
    "applicationId": "d8c3a66a-e6f9-449e-8a54-2416e2d61aec",
    "authenticationKey": "XXXXXXXXXXXXXXXXX",
    "description": "This is test description",
    "directoryId": "ff4e2442-65ab-4dc2-9e5b-1ea02d3d94eb",
    "isGovCloud": true,
    "name": "My Azure Connector",
    "remediationEnabled":true,
    "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "pollingFrequency": {
        "hours": 4,
        "minutes": 0
    }
}
```

**Response**

```json
{
    "applicationId": "d8c3a66a-e6f9-449e-8a54-2416e2d61aec",
```
"connectorId": "674292e0-5223-11e9-be90-4dfe52eda963",
"description": "This is test description",
"directoryId": "ff4e2442-65ab-4dc2-9e5b-1ea02d3d94eb",
"error": "string",
"groups": [],
"isGovCloud": true,
"lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
"name": "My Azure Connector",
"provider": "AZURE",
"remediationEnabled": true,
"state": "PENDING",
"subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
"totalAssets": 0,
"pollingFrequency": {
  "hours": 4,
  "minutes": 0
},
"nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020"
Run Connector (Azure)

/rest/v1/azure/connectors/run

[POST]

Specify the IDs of the connectors that you want to run.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| connectorRunRequest     | (Array[string]) Specify the IDs of the connector that you want to execute/run.
|                         | Example:                                                                    |
|                         | ["string"]                                                                |

Sample - Get details of a specific connectors in user's account

API request

```
curl -k -X GET -u <username>:<password> -d '"67a51d30-14d4-12e9-aae4-31d950d53bd8"]'
'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors`
```

Response

No Content
Response Code: 204
Update Connector (Azure)

/rest/v1/azure/connectors/{connectorId}

[PUT]

Specify the connector ID and the details of the connector that you would want to update in the connectorBody parameter. Your connector details get updated.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(string) Specify the unique Id associated with connector in the user's scope.</td>
</tr>
<tr>
<td>connectorBody</td>
<td>(body) Specify only those connector details that you want to update. Refer to the following example for syntax.</td>
</tr>
</tbody>
</table>

```json
{
   "applicationId": "string",
   "authenticationKey": "string",
   "description": "string",
   "directoryId": "string",
   "isGovCloud": true,
   "name": "string",
   "remediationEnabled": true,
   "pollingFrequency": {
      "hours": 0,
      "minutes": 0
   }
}
```

Where,

- applicationId: Unique identifier of the application you
create on Azure portal.

-authenticationKey: The secret key generated after you provide permission to the application to access the Windows Azure Service.

description is optional and you can give a short description stating the purpose of the connector you want to create.

directoryId: Unique identifier of your Azure Active Directory.

-isGovCloud (boolean): A flag indicating whether the Connector also is created GovCloud region or not. Set this flag to true to create the connector for GovCloud.

-name is the name for the connector you want to update.

-remediationEnabled (boolean). A flag to enable or disable remediation for the connector. To enable remediation, the remediationEnabled flag needs to be set to true. This flag is optional. By default, this flag is configured to false. If remediation is enabled on a connector, the response includes "remediationEnabled": true. If you do not set the flag, the existing remediation status of the connector is displayed in the response.

Note: The remediation feature is available only to Cloud Security Assessment (CSA) subscribers and is enabled by default.

- pollingFrequency: Polling frequency for a connector decides the rate at which the connector should poll the cloud provider and fetch the data.

-hours: Specify the time in hours. The valid range is 1 to 24 hours.

-minutes: Specify the time in minutes. The valid range is 0 to 59 minutes.
You can configure frequency from minimum one hour to maximum 24 hours. We recommend that you configure frequency of 4 hours or more for optimal use of your connector. Configuring a low polling frequency (lesser than 4 hours) can affect the performance of the connector and may result in Microsoft Azure API throttling error.

Note: Configuration of connector polling frequency is enabled only for Cloud Security Assessment (CSA) users. For all other users, the default connector polling frequency is pre-configured and even if you update the connector polling frequency, the change in frequency will not reflect. For Cloud Inventory (CI) users, the pre-configured frequency is 24 hours. For the trial period, the pre-configured frequency it is 4 hours.

Sample - Update Azure Connector

Update connector in the user’s scope.

**API request**

```
curl -k -X GET -u <username>:<password>
{
  "applicationId": "d8c3a66a-e6f9-449e-8a54-2416e2d61aec",
  "authenticationKey": "XXXXXXXXXXXXXXXX",
  "description": "Test description updated",
  "directoryId": "ff4e2442-65ab-4dc2-9e5b-1ea02d3d94eb",
  "isGovCloud": true,
  "name": "My Azure Connector",
  "remediationEnabled=true",
  "pollingFrequency": {
    "hours": 6,
    "minutes": 0
  }
}
'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors/7d80a840-40b6-11e9-9078-111111111111'
```

**Response**

```
{
  "name": "My Azure Connector",
```
Enable Connector (Azure)

/rest/v1/azure/connectors/connectors/enable

[PATCH]

We give you the flexibility to enable or disable a connector. By default, all connectors you create are in enabled state. When you disable a connector, it is not eligible for auto-run or manual run. You can enable a disabled connector to make it eligible for auto-run or manual run. You can view connector details, update or delete a disabled connector.

Note: If a combination of connectors (enabled/disabled) or only disabled connector are run using Run API request call, the response is always acknowledged: true. The Run connector API always skips disabled connector for all clouds.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user's scope to be enabled.</td>
</tr>
</tbody>
</table>

Sample - Enable Azure Connector

Let us consider an example to enable an Azure connector.

**API request**

```
curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors/connectors/enable'-d "[ "359dba68-8559-30f4-88d5-d4720647a23f"]"
```

**Response**

```json
{
  "acknowledged": true
}
```
Disable Connector (Azure)

/rest/v1/azure/connectors/connectors/disable

[PATCH]

We give you the flexibility to enable or disable a connector. By default, all connectors you create are in enabled state. When you disable a connector, it is not eligible for auto-run or manual run. You can enable a disabled connector to make it eligible for auto-run or manual run. You can view connector details, update or delete a disabled connector.

Note: If a combination of connectors (enabled/disabled) or only disabled connector are run using Run API request call, the response is always acknowledged: true. The Run connector API always skips disabled connector for all clouds.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user's scope to be disabled.</td>
</tr>
</tbody>
</table>

Sample - Disable Azure Connector

Let us consider an example to disable an Azure connector.

**API request**

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors/connectors/disable' -d "[ "359dba68-8559-30f4-88d5-d4720647a23f"]"

**Response**

```json
{
  "acknowledged": true
}
```
CloudView APIs

}

Delete Connector

/rest/v1/azure/connectors

[DELETE]

Delete the specified connector which is in the user’s scope.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(Array[string]) Specify the unique Id associated with connector in the user’s scope.</td>
</tr>
<tr>
<td></td>
<td>Example: [&quot;67a51d30-14d4-12e9-aae4-31d950d53bd7&quot;]</td>
</tr>
</tbody>
</table>

Sample - Delete Azure Connector in user's account

**API request**

curl -X DELETE -u <username>:<password>  
-d '[[215947d0-569d-11e9-8032-2fa7ed9d9b64]]'  
'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors'

**Response**

No Content  
Response Code: 204
Azure Evaluations

Get the stats for specified control id and resource id

/\texttt{rest/v1/azure/evaluations/stats/\{controlId\}/\{connectorId\}?resourceId=\{resourceId\}}

[GET]

Specify the details such as control ID, resource ID, and connector ID to get the statistics for specified control and resource ID.

\textbf{Note}: By default, the response includes the data for last 24 hours.

\textbf{Input Parameters}

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>controlId</td>
<td>(string) Specify the control ID of a control for which resources evaluated need to be fetched</td>
</tr>
<tr>
<td>resourceId</td>
<td>(string) Specify the unique ID of the resource being evaluated.</td>
</tr>
<tr>
<td>connectorId</td>
<td>(string) Specify the unique Id associated with the connector in the user’s scope.</td>
</tr>
</tbody>
</table>

\textbf{Sample - Get list of connectors in user’s account}

Return the list of all connectors in the user’s scope.

\textbf{API request}

```
```
Response

{
    "firstEvaluated": "2021-03-08T23:01:10+0000",
    "lastEvaluated": "2021-03-22T12:07:05+0000",
    "dateReopen": "2021-03-25T08:57:17+0000",
    "dateFixed": "2021-03-25T08:51:45+0000",
}
Get the list of evaluations as per the subscription for Azure Controls

/\rest/v1/azure/evaluations/{subscriptionId}

[GET]

Specify the details such as control ID, resource ID, and connector ID to get the statistics for specified control and resource ID.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriptionId</td>
<td>(string) Specify the unique Id associated with your Azure subscription.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter the resources list by providing a query using Qualys syntax.</td>
</tr>
</tbody>
</table>

If you do not add a date filter, by default the data for last 24 hours is included in the response. If you need data for specific date or date range, form your filter query using evaluatedOn token.

Examples:

Show resources discovered within certain dates

```evaluatedOn: [2019-01-01 ... 2019-03-01]```

Show resources updated starting 2019-01-01, ending 1 month ago

```evaluatedOn: [2019-01-01 ... now-1m]```

Show resources updated starting 2 weeks ago, ending 1 second ago

```evaluatedOn: [now-2w ... now-1s]```

Show resources discovered on specific date

```evaluatedOn: 2019-01-08```
API request

```
curl -k -X GET -u <username>:<password>
```

Response

```
{
  "content": [
    {
      "controlName": "Ensure that Adaptive Application Controls is set to On",
      "policyName": "CIS Microsoft Azure Foundations Benchmark",
      "criticality": "HIGH",
      "service": "SECURITY_CENTER",
      "result": "PASS",
      "controlId": "50003",
      "passedResources": 1,
      "failedResources": 0,
      "passWithExceptionResources": 1
    },
    ...
  ],
  "pageable": {
    "sort": {
      "unsorted": true,
      "sorted": false
    },
    "pageSize": 100,
    "pageNumber": 0,
    "offset": 0,
    "unpaged": false,
    "paged": true
  },
  "totalElements": 16,
  "totalPages": 1,
  "last": true,
  "first": true,
  "sort": {
    "unsorted": true,
    "sorted": false
  },
  "numberOfElements": 16,
  "size": 100,
  "number": 0
}
```
Get the resources evaluated for the specified Azure subscription and control id

/rest/v1/azure/evaluations/{subscriptionId}/resources/{controlId}

[GET]

Specify the details such as control ID, resource ID, and connector ID to get the statistics for specified control and resource ID.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriptionId</td>
<td>(string) Specify the unique Id associated with your Azure subscription.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter the resources list by providing a query using Qualys syntax.</td>
</tr>
<tr>
<td></td>
<td>If you do not add a date filter, by default the data for last 24 hours is</td>
</tr>
<tr>
<td></td>
<td>included in the response. If you need data for specific date or date range,</td>
</tr>
<tr>
<td></td>
<td>form your filter query using evaluatedOn token.</td>
</tr>
</tbody>
</table>

Examples:

Show resources discovered within certain dates
evaluatedOn: [2019-01-01 ... 2019-03-01]

Show resources updated starting 2019-01-01, ending 1 month ago
evaluatedOn: [2019-01-01 ... now-1m]

Show resources updated starting 2 weeks ago, ending 1 second ago
evaluatedOn: [now-2w ... now-1s]

Show resources discovered on specific date
evaluatedOn: 2019-01-08
**controlId**  (string) Specify the control ID of a control for which resources evaluated need to be fetched

**pageNo**  (integer) The page to be returned.

**pageSize**  (integer) The number of records per page to be included in the response.

**Sample - Get all or filter the evaluations for your account**

**API request**


**Response**

```
{
    "content": [
        {
            "resourceId": "TestResource",
            "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
            "evaluatedOn": "2020-03-15T07:07:24+0000",
            "evidences": [
                {
                    "settingName": "Data Encryption",
                    "actualValue": "On",
                    "expectedValue": ""
                }
            ],
            "resourceType": "SQL_SERVER_DATABASE",
            "result": "PASS",
            "evaluationDates": {
                "firstEvaluated": "2020-03-13T15:02:07+0000",
                "lastEvaluated": "2020-03-22T11:06:07+0000",
                "dateReopen": null,
                "dateFixed": null
            }
        },
        {
            "resourceId": "dnd-automation-mcheck-pass",
            "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
```
Connector Groups Management APIs

Get Groups

/rest/v1/groups

[GET]

Fetch the list of groups associated with the user.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Sample - Get the list of all groups associated with the user

**API request**

```
curl -k -X GET -u <username>:<password> 
'https://<QualysURL>/cloudview-api/rest/v1/groups?pageNo=0&pageSize=3'
```

**Response**

```
{
  "content": [
    {
      "name": "empty",
      "uuid": "bad111c0-5de8-31ef-1111-4f1c2bd1111",
      "connectors": []
    },
    {
      "name": "sample_1",
      "uuid": "0101f80-fe1d-38c5-1111-56ac16361111",
      "connectors": [
```
CloudView APIs

{
    "connectorUuid": "d0830ee0-c4b7-11e9-b442-ad0de981ea5c",
    "accountIdentifier": "XXXXXXXXXXXX",
    "cloudType": "AWS"
}

},

{
    "name": "Sample_2_Azure",
    "uuid": "ce7c0d23-af60-320f-b8cb-f6b51b2fb8a5",
    "connectors": [
        {
            "connectorUuid": "5926c280-c587-11e9-9230-31a5d0c73f76",
            "accountIdentifier": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXX",
            "cloudType": "AZURE"
        },
        {
            "connectorUuid": "2e0c1660-d061-11e9-ad71-df4fba75b3c5",
            "accountIdentifier": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXX",
            "cloudType": "AZURE"
        }
    ]
},

"pageable": {
    "sort": {
        "unsorted": true,
        "sorted": false
    },
    "pageSize": 3,
    "pageNumber": 0,
    "offset": 0,
    "unpaged": false,
    "paged": true
},

"totalElements": 27,
"last": false,
"totalPages": 9,
"first": true,
"sort": {
    "unsorted": true,
    "sorted": false
},

"numberOfElements": 3,
"size": 3,
"number": 0
Create Group

/rest/v1/groups/{cloudType}

[POST]

The groups help you to organize your connectors and to manage user access to them. You can create groups and associate it with connectors and form connector groups or segregate connectors using a specific group for a connector as well. Use groups to provide access or restrict access to users you create.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupName</td>
<td>(String) Provide a name for the group you want to create. Ensure that the name is unique.</td>
</tr>
</tbody>
</table>

Sample - Create a group in CloudView

**API request**

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/groups?groupName=sample_group'

**Response**

```
{
    "name": "sample_group",
    "groupId": "745aca99-6ab6-3e1d-8e65-7b4febc0005e"
}
```
Update Group

/rest/v1/groups/connectors

[POST]

The groups help you to organize your connectors and to manage user access to them. You can update groups and associate it with connectors and form connector groups or segregate connectors using a specific group for a connector as well. Use groups to provide access or restrict access to users you create.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorGroupModifyRequest</td>
<td>(body) Use this to specify the connector group IDs that you want to add and remove groups.</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "accountIdentifiers": ["string"],
  "add": {
    "groupIds": ["string"]
  },
  "remove": {
    "groupIds": ["string"]
  }
}
```

where,

accountIdentifiers: The unique identifier associated with a connector. For every cloud provider, the identifier is different.

- AWS: account ID (Example: 11111111111)
- Azure: subscription ID (Example: 11111111111111111111111111111111)
CloudView APIs

<table>
<thead>
<tr>
<th>cloudType</th>
<th>Select the cloud provider to which the connector being updated belongs.</th>
</tr>
</thead>
</table>

Sample - Update Connectors in a group

**API request**

```bash
curl -k -X POST -u <username>:<password> \\
'https://<QualysURL>/cloudview-api/rest/v1/groups/connectors?cloudType=GCP'
```

**Request POST Data**

```json
{
  "accountIdentifiers": [ "gcp-demo" ],
  "add":
  {
    "groupIds": [ "ea4b240f-c27c-30a6-ba28-8fc9a38fa8d1" ]
  },
  "remove":
  {
    "groupIds": [ "8b23977c-9f28-3007-8c11-c0469494053f" ]
  }
}'<QualysURL>/cloudview-api/rest/v1/groups/connectors?cloudType=GCP'
```

**Response**

No Content
Response Code: 204
Get Group Details

/rest/v1/groups/{groupUuid}

[GET]

You can get details of a group by specifying the unique Id associated with a group.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupUuid</td>
<td>(integer) Unique Id associated with each group.</td>
</tr>
<tr>
<td></td>
<td>Example: groupUuid: b3e9036d-b546-30d4-99fb-cb64b15efffa</td>
</tr>
</tbody>
</table>

Sample - Get the list of groups

**API request**

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/groups/2d16e55b-d01a-3fcb-8239-3d4ea04bc54a'

**Response**

```json
{
    "name": "group2",
    "uuid": "2d16e55b-d01a-3fcb-8239-3d4ea04bc54a",
    "connectors": [
        {
            "connectorUuid": "66e7d1f0-c8c2-11e9-9fcb-85661d3ad949",
            "accountIdentifier": "gcp-demo",
            "cloudType": "GCP"
        },
        {
            "connectorUuid": "271bdec0-d2f1-11e9-93db-85a6f54e372e",
            "accountIdentifier": "gcp_example_2",
            "cloudType": "GCP"
        }
    ]
}```
CloudView APIs

} ]
}

Control Meta Data

Get Control Metadata

/rest/v1/controls/metadata/list

[GET]

Fetch the metadata for controls. You can use the filters we support to narrow down the controls to be fetched in the response. We support two method: XML and JSON for the response. You can choose the required method and fetch the data in XML and JSON.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>(query) Filter the controls list by providing a query using Qualys syntax. You can use the following tokens:</td>
</tr>
<tr>
<td></td>
<td>- control.name</td>
</tr>
<tr>
<td></td>
<td>- resource.type</td>
</tr>
<tr>
<td></td>
<td>- service.type</td>
</tr>
<tr>
<td></td>
<td>- cid</td>
</tr>
<tr>
<td></td>
<td>- provider</td>
</tr>
<tr>
<td></td>
<td>- control.criticality</td>
</tr>
<tr>
<td></td>
<td>- control.type</td>
</tr>
<tr>
<td></td>
<td>- policy.name</td>
</tr>
<tr>
<td></td>
<td>- createdDate</td>
</tr>
<tr>
<td></td>
<td>- modifiedDate</td>
</tr>
<tr>
<td></td>
<td>- isCustomizable</td>
</tr>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Sample - Get the control metadata (AWS)
**API request**

curl -X GET -u <username>:<password>
'https://<QualysURL>/cloudview-api/rest/v1/controls/metadata/list?filter=provider%3AAWS&pageNo=0&pageSize=100' -H "accept: application/xml"

---

**Response (XML)**

```xml
<?xml version='1.0' encoding='UTF-8'?>
<CONTROL_LIST_OUTPUT>
    <DATETIME>2021-07-06T12:50:34.526+00:00</DATETIME>
    <CONTROL_LIST>
        <CONTROL>
        <CID>1</CID>
        <CONTROL_NAME>Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password</CONTROL_NAME>
        <CREATED>2020-05-07T12:56:56+0000</CREATED>
        <MODIFIED>2021-05-06T11:31:00+0000</MODIFIED>
        <CONTROL_TYPE>System Defined</CONTROL_TYPE>
        <PROVIDER>AWS</PROVIDER>
        <IS_CUSTOMIZABLE>false</IS_CUSTOMIZABLE>
        <SERVICE_TYPE><![CDATA[IAM]]></SERVICE_TYPE>
        <CRITICALITY>HIGH</CRITICALITY>
        <EVALUATION>
            <EVALUATION_DESCRIPTION><![CDATA[<p>Check IAM Users having console password enabled has MFA Set to True.</p>
/n
Changes in account credentials may take upto 4 hours to get reflected in the AWS IAM evaluations. The time taken depends on when the last credential report was fetched by the Cloud View service and the time when changes were made in AWS IAM]]></EVALUATION_DESCRIPTION>
            <PASS_MESSAGE>IAM user is configured with MFA.</PASS_MESSAGE>
            <FAIL_MESSAGE>IAM user is not configured with MFA.</FAIL_MESSAGE>
        </EVALUATION>
    </CONTROL>
    </CONTROL_LIST>
</CONTROL_LIST_OUTPUT>
```

---

Multi-Factor Authentication (MFA) adds an extra layer of protection on top of a user name and password. With MFA enabled,
when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device. It is recommended that MFA be enabled for all accounts that have a console password.

Response (JSON)

```
{
    "dateTime":"2021-07-06T12:52:15.637+00:00",
    "control": [
        {
            "cid": 1,
            "controlName": "Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password",
            "created": "2020-05-07T12:56:56+0000",
            "modified": "2021-05-06T11:31:00+0000",
            "controlType": "System Defined",
            "provider": "AWS",
            "isCustomizable": false,
            "serviceType": "IAM",
            "criticality": "HIGH",
            "evaluation": {
                "evaluationDescription": "Check IAM Users having console password enabled has MFA Set to True."
            },
            "passMessage": "IAM user is configured with MFA.",
            "failMessage": "IAM user is not configured with MFA."
        }
    ]
}```
Multi-Factor Authentication (MFA) adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device. It is recommended that MFA be enabled for all accounts that have a console password.

CIS reference: CIS Amazon Web Services Foundations Benchmark v1.3.0 - 08-07-2020: Recommendation #1.10

Sample - Get the control metadata (Azure)

API request

curl -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors?pageNo=0&pageSize=50'

Response (XML)

<?xml version='1.0' encoding='UTF-8'?>
<CONTROL_LIST_OUTPUT>
  <DATETIME>2021-07-06T12:55:48.065+00:00</DATETIME>
  <CONTROL_LIST>
    <CONTROL>
      <CID>50001</CID>
      <CONTROL_NAME>Ensure that Data encryption is set to ON for a SQL database</CONTROL_NAME>
      <CREATED>2020-05-07T01:27:53+0000</CREATED>
      <MODIFIED>2021-04-22T06:41:05+0000</MODIFIED>
      <CONTROL_TYPE>System Defined</CONTROL_TYPE>
      <PROVIDER>AZURE</PROVIDER>
      <IS_CUSTOMIZABLE>false</IS_CUSTOMIZABLE>
      <SERVICE_TYPE><![CDATA[Azure SQL]]></SERVICE_TYPE>
      <CRITICALITY>HIGH</CRITICALITY>
      <EVALUATION>
        <EVALUATION_DESCRIPTION><![CDATA[This control ensures that `Transparent Data Encryption' is enabled for a threat detection policy on a SQL server.]]></EVALUATION_DESCRIPTION>
      </EVALUATION>
    </CONTROL>
  </CONTROL_LIST>
</CONTROL_LIST_OUTPUT>
Response (JSON)

```xml
<?xml version='1.0' encoding='UTF-8'?>
<CONTROL_LIST_OUTPUT>
  <DATETIME>2021-07-06T12:55:48.065+00:00</DATETIME>
  <CONTROL_LIST>
    <CONTROL>
      <CID>50001</CID>
      <CONTROL_NAME>Ensure that Data encryption is set to ON for a SQL database</CONTROL_NAME>
      <CREATED>2020-05-07T01:27:53+0000</CREATED>
      <MODIFIED>2021-04-22T06:41:05+0000</MODIFIED>
      <CONTROL_TYPE>System Defined</CONTROL_TYPE>
      <PROVIDER>AZURE</PROVIDER>
      <IS_CUSTOMIZABLE>false</IS_CUSTOMIZABLE>
      <SERVICE_TYPE><![CDATA[Azure SQL]]></SERVICE_TYPE>
      <CRITICALITY>HIGH</CRITICALITY>
      <EVALUATION>
        <EVALUATION_DESCRIPTION><![CDATA[This control ensures that `Transparent Data Encryption' is enabled for a threat detection policy on a SQL server.]]></EVALUATION_DESCRIPTION>
        <PASS_MESSAGE>Transparent Encryption is Enabled for a SQL Database</PASS_MESSAGE>
        <FAIL_MESSAGE>Transparent Encryption is not Enabled for a SQL Database</FAIL_MESSAGE>
      </EVALUATION>
      <SPECIFICATION><![CDATA[Enable Transparent Data Encryption on every SQL database.]]></SPECIFICATION>
    </CONTROL>
  </CONTROL_LIST>
</CONTROL_LIST_OUTPUT>
```
Sample - Get the control metadata (GCP)

**API request**
curl -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors?pageNo=0&pageSize=50'

**Response (XML)**
```xml
<?xml version='1.0' encoding='UTF-8'?>
<CONTROL_LIST_OUTPUT>
  <DATETIME>2021-07-06T12:57:27.547+00:00</DATETIME>
  <CONTROL_LIST>
    <CONTROL>
      <CID>52000</CID>
      <CONTROL_NAME>Ensure that corporate login credentials are used instead of Gmail accounts</CONTROL_NAME>
      <CREATED>2020-05-07T01:24:08+0000</CREATED>
      <MODIFIED>2021-05-19T09:00:54+0000</MODIFIED>
      <CONTROL_TYPE>System Defined</CONTROL_TYPE>
      <PROVIDER>GCP</PROVIDER>
      <IS_CUSTOMIZABLE>false</IS_CUSTOMIZABLE>
      <SERVICE_TYPE><![CDATA[IAM & Admin]]></SERVICE_TYPE>
      <CRITICALITY>MEDIUM</CRITICALITY>
      <EVALUATION>
        <EVALUATION_DESCRIPTION><![CDATA[
This control ensures that corporate login credentials are used instead of Gmail accounts.
]]>
      </EVALUATION>
      <POLICY_NAME_List>
        <POLICY_NAME>CIS Google Cloud Platform Foundation Benchmark</POLICY_NAME>
      </POLICY_NAME_List>
    </CONTROL>
  </CONTROL_LIST>
</CONTROL_LIST_OUTPUT>
```
Response (JSON)

```
{
    "dateTime": "2021-07-06T12:58:24.633+00:00",
    "control": [
        {
            "cid": 52000,
            "controlName": "Ensure that corporate login credentials are used instead of Gmail accounts",
            "created": "2020-05-07T01:24:08+0000",
            "modified": "2021-05-19T09:00:54+0000",
            "controlType": "System Defined",
            "provider": "GCP",
            "isCustomizable": false,
            "serviceType": "IAM & Admin",
            "criticality": "MEDIUM",
            "evaluation": {
                "evaluationDescription": "This control ensures that corporate login credentials are used instead of Gmail accounts.

                "passMessage": "Corporate login credentials are used instead of Gmail account",
                "failMessage": "Corporate login credentials are not used instead of Gmail account",
                "evaluationCriteria": []
            },
            "specification": "Use corporate login credentials instead of Gmail accounts.

            CIS reference: Google Cloud Platform Foundation Benchmark v1.1.0 - 03-12-2020: Recommendation #1.1"
        }
    ],
    "policyNames": [
        "CIS Google Cloud Platform Foundation Benchmark"
    ]
}
```

DTD (metadata.dtd)

Below the metadata.dtd used in the new API that we introduced.

```xml
<!DOCTYPE CONTROL_LIST_OUTPUT [
    <!ELEMENT CONTROL_LIST_OUTPUT (
        DATETIME, WARNING?, CONTROL_LIST)*>
]>
```
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT WARNING (CODE,TEXT,URL)>*
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT CONTROL_LIST (CONTROL)>*
<!ELEMENT CONTROL (CID,CONTROL_NAME,CREATED,MODIFIED,CONTROL_TYPE,PROVIDER,IS_CUSTOMIZABLE,SERVICE_TYPE,CRITICALITY,EVALUATION,SPECIFICATION,RATIONALE,MANUAL_REMEDIATION,REFERENCES,RESOURCE_TYPE,REMEDIATION_ENABLED,POLICY_NAME_LIST)>*
<!ELEMENT CID (#PCDATA)>
<!ELEMENT CONTROL_NAME (#PCDATA)>
<!ELEMENT CREATED (#PCDATA)>
<!ELEMENT MODIFIED (#PCDATA)>
<!ELEMENT CONTROL_TYPE (#PCDATA)>
<!ELEMENT PROVIDER (#PCDATA)>
<!ELEMENT IS_CUSTOMIZABLE (#PCDATA)>
<!ELEMENT SERVICE_TYPE (#PCDATA)>
<!ELEMENT CRITICALITY (#PCDATA)>
<!ELEMENT EVALUATION (EVALUATION_DESCRIPTION,PASS_MESSAGE,FAIL_MESSAGE,EVALUATION_CRITERIA_LIST)>*
<!ELEMENT EVALUATION_DESCRIPTION (#PCDATA)>
<!ELEMENT PASS_MESSAGE (#PCDATA)>
<!ELEMENT FAIL_MESSAGE (#PCDATA)>
<!ELEMENT EVALUATION_CRITERIA_LIST (EVALUATION_CRITERIA)>*
<!ELEMENT EVALUATION_CRITERIA (LABEL,OPERATOR,VALUE)>*
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT OPERATOR (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!ELEMENT SPECIFICATION (#PCDATA)>
<!ELEMENT RATIONALE (#PCDATA)>
<!ELEMENT MANUAL_REMEDIATION (#PCDATA)>
<!ELEMENT REFERENCES (#PCDATA)>
<!ELEMENT RESOURCE_TYPE (#PCDATA)>
<!ELEMENT REMEDIATION_ENABLED (#PCDATA)>
<!ELEMENT POLICY_NAME_LIST (POLICY_NAME)>*
<!ELEMENT POLICY_NAME (#PCDATA)>
GCP APIs

GCP Connector

We support the following operations for GCP Connector.

- Get list of connectors
- Get the details of a connector
- Create a new connector
- Run the provided connector
- Update the existing connector
- Enable Connector (GCP)
- Disable Connector (GCP)
- Delete the provided connectors

GCP Evaluations

We support the following control evaluations for GCP resources

- Get the stats for specified control id and resource id
- Get the list of evaluations per account for GCP controls
- Get the resources evaluated for the specified GCP account and control id
## Get GCP Connectors

/rest/v1/gcp/connectors

[GET]

List all GCP connectors in the user’s account.

### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| filter     | Filter the connectors list by providing a query using Qualys syntax. The following search tokens are supported.  
- name: Name of the connector  
- description: Short description of the connector  
- state: Connector status The valid values are SUCCESS, PENDING, REGIONS_DISCOVERED, ERROR  
- connector.uuid: Unique Id assigned to the connector. For example, 6192ce15-e790-3fe2-a02c-b4bc75ecf123"  
- lastSyncedOn: Date and time when the connector synced with the cloud provider. Note: This time should be in UTC time |
| pageNo     | (integer) The page to be returned. |
| pageSize   | (integer) The number of records per page to be included in the response. |
| sort       | (keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc. |
Sample - Get list of GCP connectors in user's account

Return the list of all GCP connectors in the user's scope.

API request

```
curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors?pageNo=0&pageSize=50'
```

Response

```
{
    "content": [
        {
            "name": "GCP Connector",
            "connectorId": "57d5b7c1-685a-3da8-ab11-c465c0ee60e6",
            "description": "",
            "provider": "GCP",
            "state": "SUCCESS",
            "totalAssets": 238,
            "lastSyncedOn": "Thu May 20 11:51:41 UTC 2021",
            "nextSyncedOn": "Thu May 20 15:26:27 UTC 2021",
            "remediationEnabled": true,
            "isDisabled": false,
            "projectId": "my-project-XXXXXXXXXX",
            "groups": [
                {
                    "name": "SAMPLe Group",
                    "uuid": "4659b745-c06d-39ef-8bc1-e6f7f5acee12"
                },
                {
                    "name": "GCP Connectors",
                    "uuid": "173236f5-5b34-3757-9782-5e9dec1c1709"
                }
            ],
            "pollingFrequency": {
                "hours": 4,
                "minutes": 0
            }
        }
    ]
}
```
Sample - Filter the list of GCP connectors in success state and sort in descending order with lastSyncedOn

**API request**

curl -k -X GET -u <username>:<password> "https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors?filter=state%3DSUCCESS&pageNo=1&pageSize=50&sort=lastSyncedOn%3Adesc"

**Response**

```json
"state": "SUCCESS",
"totalAssets": 146,
"lastSyncedOn": "Thu May 20 11:51:36 UTC 2021",
"nextSyncedOn": "Thu May 20 14:58:36 UTC 2021",
"remediationEnabled": true,
"isDisabled": false,
"projectId": my-project-XXXXXXXXXX",
...
"pageable": {
   "sort": {
      "unsorted": true,
      "sorted": false
   },
   "pageSize": 50,
   "pageNumber": 0,
   "offset": 0,
   "paged": true,
   "unpaged": false
},
"totalElements": 2,
"last": true,
"totalPages": 1,
"first": true,
"sort": {
   "unsorted": true,
   "sorted": false
},
"numberOfElements": 2,
"size": 50,
"number": 0
```

95
"content": [
  {
    "name": "Sample GCP Connector",
    "connectorId": "a7c152e6-2417-3bbf-980d-bfe6f5912345",
    "description": "gcp connector",
    "provider": "GCP",
    "state": "SUCCESS",
    "totalAssets": 67,
    "lastSyncedOn": "Thu Nov 26 07:50:27 UTC 2020",
    "nextSyncedOn": "Thu Nov 26 11:50:27 UTC 2020",
    "isDisabled": false,
    "projectId": "project1",
    "groups": [],
    "pollingFrequency": {
      "hours": 4,
      "minutes": 0
    }
  },
  ...

"numberOfElements": 2,
  "size": 50,
  "number": 0
]
Get GCP Connector Details

/rest/v1/gcp/connectors/{connectorId}

[GET]

View details for a specific GCP connector which is in the user’s scope.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(string) Specify the unique Id associated with connector in the user’s scope.</td>
</tr>
</tbody>
</table>

Sample - Get details of a specific GCP connector in user’s account

**API request**

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors/1111a111-1111-11a1-a1a1-1aa1a1111111'

**Response**

```json
{
    "name": "GCP Connector",
    "connectorId": "1111a111-1111-11a1-a1a1-1aa1a1111111",
    "description": "",
    "provider": "GCP",
    "state": "SUCCESS",
    "totalAssets": 238,
    "lastSyncedOn": "Thu May 20 11:51:41 UTC 2021",
    "nextSyncedOn": "Thu May 20 15:26:27 UTC 2021",
    "remediationEnabled": true,
    "isDisabled": false,
    "projectId": "my-project-xxxxxxxxxxxxx",
    "groups": [
        {
            "name": "GCP Connectors",
            "uuid": "173236f5-5b34-3757-9782-5e9dec1c1709"
        }
    ]
}
{  
  "name": "Sample Group",
  "uuid": "4659b745-c06d-39ef-8bc1-e6f7f5acee12"
}
,
  "pollingFrequency": {
    "hours": 4,
    "minutes": 0
  }
}
Create GCP Connector

/rest/v1/gcp/connectors

[POST]

Specify the connector details such as name, description, polling frequency, project ID and upload the configuration (JSON) file and create a new connector.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the connector</td>
</tr>
<tr>
<td>description</td>
<td>Description of the connector</td>
</tr>
<tr>
<td>configFile</td>
<td>Provide the configuration file.</td>
</tr>
<tr>
<td>pollingFrequencyInHrs</td>
<td>Polling frequency for a connector decides the rate at which the connector should poll the cloud provider and fetch the data.</td>
</tr>
<tr>
<td>pollingFrequencyInMinutes</td>
<td>- pollingFrequencyInHrs: Specify the time in hours. The valid range is 1 to 24 hours.</td>
</tr>
<tr>
<td></td>
<td>- pollingFrequencyInMinutes: Specify the time in minutes. The valid range is 0 to 59 minutes.</td>
</tr>
</tbody>
</table>

You can configure frequency from minimum one hour to maximum 24 hours. We recommend that you configure frequency of 4 hours or more for optimal use of your connector. Configuring a low polling frequency (lesser than 4 hours) can affect the performance of the connector and may result in AWS API throttling error.

Note: Configuration of connector polling frequency is enabled only for Cloud Security
CloudView APIs

Assessment (CSA) users. For all other users, the default connector polling frequency is pre-configured and even if you update the connector polling frequency, the change in frequency will not reflect. For Cloud Inventory (CI) users, the pre-configured frequency is 24 hours. For the trial period, the pre-configured frequency it is 4 hours.

**projectId**

Provide the project Id for the GCP connector you want to create.

**remediationEnabled:** {true, false} (boolean). A flag to enable or disable remediation for the connector. To enable remediation, the remediationEnabled flag needs to be set to true. This flag is optional. By default, this flag is configured to false. If remediation is enabled on a connector, the response includes "remediationEnabled": true. If you do not set the flag, the existing remediation status of the connector is displayed in the response.

Note: The remediation feature is available only to Cloud Security Assessment (CSA) subscribers and is enabled by default.

Sample - Create a new GCP Connector

**API request**

```
curl -k -X POST -d <username>:<password>
-H "Content-Type: multipart/form-data"
-F "configFile=@sample_project-208707-56baacf5cc17.json;type=application/json" -F "description=sample description" -F "name=Sample GCP Connector" -F "pollingFrequencyInHrs=10" -F "pollingFrequencyInMinutes=15" -F "projectId=sample_project-208707" -H "remediationEnabled=true" 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors'
```

Note: Upload the configuration file required for GCP connector creation.

**Response**

100
"name": "Sample GCP Connector",
"connectorId": "fde424cb-5b3c-3ccd-b024-b40613cf1e51",
"description": "sample description",
"provider": "GCP",
"state": "PENDING",
"totalAssets": 0,
"lastSyncedOn": "Thu Feb 25 03:43:08 UTC 2021",
"nextSyncedOn": "Thu Feb 25 13:58:08 UTC 2021",
"remediationEnabled": true,
"isDisabled": false,
"projectId": "sample_project-208707",
"groups": [],
"pollingFrequency": { "hours": 10, "minutes": 15 }
}
Run Connector (GCP)

/rest/v1/gcp/connectors/run

[POST]

Specify the connector details and run the specified GCP connector.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorRunRequest</td>
<td>(Array [string]) Specify the unique Id associated with connector in the user's scope.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>[</td>
</tr>
<tr>
<td></td>
<td>&quot;string&quot;</td>
</tr>
<tr>
<td></td>
<td>]</td>
</tr>
</tbody>
</table>

Sample - Run the specified GCP connector

**API request**

```bash
curl -k -X POST -d <username>:<password> -d "['1111a111-1111-11a1-a1a1-1aa1a1111111']"
'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors/run'
```

**Response**

No Content
Response Code: 204
Update Connector (GCP)

/rest/v1/gcp/connectors/{connectorId}

[PUT]

Specify the connector ID and the details to be updated to update details of the specified connector.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user’s scope.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the connector</td>
</tr>
<tr>
<td>description</td>
<td>Description of the connector</td>
</tr>
<tr>
<td>remediationEnabled: {true, false}</td>
<td>(boolean). A flag to enable or disable remediation for the connector. To enable remediation, the remediationEnabled flag needs to be set to true. This flag is optional. By default, this flag is configured to false. If remediation is enabled on a connector, the response includes &quot;remediationEnabled&quot;: true. If you do not set the flag, the existing remediation status of the connector is displayed in the response.</td>
</tr>
<tr>
<td>configFile</td>
<td>Provide the configuration file.</td>
</tr>
</tbody>
</table>

Note: The remediation feature is available only to Cloud Security Assessment (CSA) subscribers and is enabled by default.

Sample - Update the GCP connector

You can update either one or multiple elements of the GCP connector.
API request

```bash
curl -k -X PUT -u <username>:<password> -F name=Change_name_GCP -F "pollingFrequencyInHrs=4" -F "pollingFrequencyInMinutes=0" -H "remediationEnabled=true" 'https://<QualysURL/cloudview-api/rest/v1/gcp/connectors/1f325070-5152-11e9-8cf3-e3dcac181204'
```

Response

```json
{
    "name": "Change_name_GCP",
    "description": "Test description",
    "state": "PENDING",
    "totalAssets": 0,
    "projectId": "my-project-1111111111111",
    "provider": "GCP",
    "remediationEnabled": true,
    "connectorId": "1f325070-5152-11e9-8cf3-e3dcac181204",
    "lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
    "pollingFrequency": {
        "hours": 4,
        "minutes": 0
    },
    "nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020"
}
```
Enable Connector (GCP)

/rest/v1/gcp/connectors/connectors/enable

[PATCH]

We give you the flexibility to enable or disable a connector. By default, all connectors you create are in enabled state. When you disable a connector, it is not eligible for auto-run or manual run. You can enable a disabled connector to make it eligible for auto-run or manual run. You can view connector details, update or delete a disabled connector.

Note: If a combination of connectors (enabled/disabled) or only disabled connector are run using Run API request call, the response is always acknowledged: true. The Run connector API always skips disabled connector for all clouds.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user's scope to be enabled.</td>
</tr>
</tbody>
</table>

Sample - Enable GCP Connector

Let us consider an example to enable an GCP connector.

**API request**

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors/connectors/enable' -d "[ \"359dba68-8559-30f4-88d5-d4720647a23f\"]"

**Response**

```json
{
   "acknowledged": true
}
```
Disable Connector (GCP)

/rest/v1/aws/connectors/connectors/disable

[PATCH]

We give you the flexibility to enable or disable a connector. By default, all connectors you create are in enabled state. When you disable a connector, it is not eligible for auto-run or manual run. You can enable a disabled connector to make it eligible for auto-run or manual run. You can view connector details, update or delete a disabled connector.

Note: If a combination of connectors (enabled/disabled) or only disabled connector are run using Run API request call, the response is always acknowledged: true. The Run connector API always skips disabled connector for all clouds.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(integer) Specify the unique Id associated with connector in the user’s scope to be disabled.</td>
</tr>
</tbody>
</table>

Sample - Disable GCP Connector

Let us consider an example to disable an GCP connector.

**API request**

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors/connectors/disable'-d "[ "$359dba68-8559-30f4-88d5-d4720647a23f"]"

**Response**

```json
{
    "acknowledged": true
}
```
CloudView APIs

}
Delete Connector (GCP)

/rest/v1/gcp/connectors

[DELETE]

Delete the specified connector which is in the user’s scope.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorId</td>
<td>(Array [string]) Specify the unique Id associated with connector in the user’s scope.</td>
</tr>
</tbody>
</table>

Sample - Delete a specific GCP connectors in user’s account

**API request**

```
curl -k -X DELETE -u <username>:<password> -d '{"1d767489-da0c-4948-a285-bf2c708c0585"}' 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors'
```

**Response**

No Content
Response Code: 204
GCP Evaluations

Get the stats for specified control id and resource id

/rest/v1/gcp/ evaluations/stats/{controlId}/{connectorId}?resourceId={resourceId}

[GET]

Specify the details such as control ID, resource ID, and connector ID to get the statistics for specified control and resource ID.

**Note:** By default, the response includes the data for last 24 hours.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>controlId</td>
<td>(string) Specify the control ID of a control for which resources evaluated need to be fetched</td>
</tr>
<tr>
<td>resourceId</td>
<td>(string) Specify the unique ID of the resource being evaluated.</td>
</tr>
<tr>
<td>connectorId</td>
<td>(string) Specify the unique Id associated with the connector in the user’s scope.</td>
</tr>
</tbody>
</table>

**Sample - Get list of connectors in user’s account**

Return the list of all connectors in the user’s scope.

**API request**

```bash
```

**Response**
{ 
  "firstEvaluated": "2021-03-08T23:01:10+0000",
  "lastEvaluated": "2021-03-22T12:07:05+0000",
  "dateReopen": "2021-03-25T08:57:17+0000",
  "dateFixed": "2021-03-25T08:51:45+0000"
}
Get the list of evaluations per project for GCP controls

/rest/v1/gcp/evaluations/{projectId}

[GET]

Specify the details such as project ID and filter details to get the list of evaluations for GCP control.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>projectId</td>
<td>(string) Specify the unique project associated with your Google Cloud Platform subscription.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter the resources list by providing a query using Qualys syntax. If you do not add a date filter, by default the data for last 24 hours is included in the response. If you need data for specific date or date range, form your filter query using evaluatedOn token. Examples:</td>
</tr>
<tr>
<td></td>
<td>Show resources discovered within certain dates evaluatedOn: [2019-01-01 ... 2019-03-01]</td>
</tr>
<tr>
<td></td>
<td>Show resources updated starting 2019-01-01, ending 1 month ago evaluatedOn: [2019-01-01 ... now-1m]</td>
</tr>
<tr>
<td></td>
<td>Show resources updated starting 2 weeks ago, ending 1 second ago evaluatedOn: [now-2w ... now-1s]</td>
</tr>
<tr>
<td></td>
<td>Show resources discovered on specific date evaluatedOn: 2019-01-08</td>
</tr>
</tbody>
</table>

**Sample - Get all or filter the evaluations for your account**
API request

```
curl -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/gcp/evaluations/my-project-1111111111111'
```

Response

```
{
  "content": [
    {
      "controlName": "Ensure that there are only GCP-managed service account keys for each service account",
      "policyName": "CIS Google Cloud Platform Foundation Benchmark",
      "criticality": "HIGH",
      "service": "IAM",
      "result": "FAIL",
      "controlId": "52001",
      "passedResources": 13,
      "failedResources": 25,
      "passWithExceptionResources": 1
    },
    ...
  ],
  "pageable": {
    "sort": {
      "sorted": false,
      "unsorted": true
    },
    "pageSize": 100,
    "pageNumber": 0,
    "offset": 0,
    "paged": true,
    "unpaged": false
  },
  "totalElements": 18,
  "last": true,
  "totalPages": 1,
  "first": true,
  "sort": {
    "sorted": false,
    "unsorted": true
  },
  "numberOfElements": 18,
  "size": 100,
  "number": 0
}
```
Get the resources evaluated for the specified GCP project and control id

/rest/v1/gcp/evaluations/{projectId}/resources/{controlId}

[GET]

Specify the details such as project Id, control Id and define your filter criteria to get the list of resources that were evaluated

**Input Parameters**

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>projectId</td>
<td>(string) Specify the project Id of a specific account in the user’s scope.</td>
</tr>
<tr>
<td>controlId</td>
<td>(string) Specify the control ID of a control for which resources evaluated need to be fetched.</td>
</tr>
</tbody>
</table>
| filter    | Filter the resources list by providing a query using Qualys syntax. If you do not add a date filter, by default the data for last 24 hours is included in the response. If you need data for specific date or date range, form your filter query using evaluatedOn token. Examples:  
  
  Show resources discovered within certain dates  
evaluatedOn: [2019-01-01 ... 2019-03-01]  

  Show resources updated starting 2019-01-01, ending 1 month ago  
evaluatedOn: [2019-01-01 ... now-1m]  

  Show resources updated starting 2 weeks ago, ending 1 second ago |
evaluatedOn: [now-2w ... now-1s]
Show resources discovered on specific date evaluatedOn: 2019-01-08

<table>
<thead>
<tr>
<th>pageNo</th>
<th>(integer) The page to be returned.</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Sample - Get the resources evaluated for the specified GCP account and control id

**API request**

```
curl -k -X GET -u <username>:<password> \\
'https://<QualysURL>/cloudview-api/rest/v1/gcp/evaluations/my-project-1111111111111/resources/50027?pageNo=0&pageSize=50'
```

**Response**

```json
{
  "content": [
    {
      "resourceId": "my-project-1111111111111",
      "region": ",",
      "projectId": "my-project-1111111111111",
      "evaluatedOn": "2020-12-15T09:15:10+0000",
      "evidences": [
        {
          "settingName": "Enable oslogin",
          "actualValue": "False"
        }
      ],
      "resourceType": "PROJECT_COMPUTE",
      "result": "PASS_WITH_EXCEPTION"
    }
  ],
  "pageable": {
    "sort": {
      "unsorted": true,
      "sorted": false
    },
    "pageSize": 50,
    "pageNumber": 0,
```

```json
```
"offset": 0,
"paged": true,
"unpaged": false
},
"last": true,
"totalPages": 1,
"totalElements": 1,
"first": true,
"sort": {
"unsorted": true,
"sorted": false
},
"numberOfElements": 1,
"size": 50,
"number": 0
Remediation

Get List of Remediation Activities

/rest/v1/remediation/activity

[GET]

Remediation allows you to select the resources you want to remediate and trigger remediation activities. All the remediation activities that you trigger from CloudView UI can be fetched. By default, the remediation activity logs are retained for 30 days. The logs older than 30 days are automatically deleted.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudType</td>
<td>Specify the cloud provider for which you want to fetch all the supported policies. You could mention AWS, AZURE, or GCP.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter the remediation activities list providing a query using Qualys syntax. The search tokens are supported are listed in the &quot;Search for Remediation Activity&quot; topic of the CloudView Online Help.</td>
</tr>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Sample - Get the list of remediation activities

Let us fetch the activities for AWS cloud provider.

API request
curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/remediation/activity?cloudType=AWS&pageNo=0&pageSize=50'

Response
{
  "content": [
  {
    "resourceId": "i-0f9ff0ee787ec1554",
    "cloudType": "AWS",
    "accountId": "XXXXXXXXXXXX",
    "region": "N. Virginia",
    "status": "Success",
    "resourceType": "Instance",
    "remediationAction": "Stop Instance",
    "triggeredBy": "John Doe",
    "remediationReason": "Sample comment"
  },
  {
    "resourceId": "cloudformation-poc-pc",
    "controlId": 60,
    "cloudType": "AWS",
    "accountId": "XXXXXXXXXXXX",
    "region": "Oregon",
    "status": "Success",
    "resourceType": "S3 Bucket",
    "remediationAction": "Control Remediation",
    "connectorName": "CLV-AWS-Connector-Remediation-116",
    "policyNames": ["CIS Amazon Web Services Foundations Benchmark", "CloudView-AWS-Test-Policy"],
    "triggeredBy": "John Doe",
    "remediationReason": "Sample comment"
  },
  ],
  "pageable": {
    "sort": {
      "field": "resourceId", "direction": "asc"
    }
  }
}
"unsorted": true,
  "sorted": false
},
  "pageSize": 50,
  "pageNumber": 0,
  "offset": 0,
  "paged": true,
  "unpaged": false
},
  "last": true,
  "totalPages": 1,
  "totalElements": 2,
  "first": true,
  "sort": {
    "unsorted": true,
    "sorted": false
  },
  "numberOfElements": 2,
  "size": 50,
  "number": 0
}
Reports

You can now generate mandate and policy based reports to get the complete picture of the compliance posture of your cloud provider account. We support report generation of policies and mandates for all the cloud providers: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

- Get Data for Specific Report
- Get List of All Supported Mandates
- Get List of All Supported Policies
- Get Report Configurations
- Get Report Details
- Create a Report
- Update a Report
- Delete Reports
Get Data for Specific Report

/rest/v1/reports/report_data/{reportId}

[GET]

Specify the report ID and you can then get the complete report.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportId</td>
<td>Unique identifier associated with every report.</td>
</tr>
</tbody>
</table>

Sample - Get the complete data of specified report

**API request**

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/reports/01164660-cfc7-11ea-a573-4395559d998e'

**Response**

```json
{
    "mandate": {
        "id": 2481,
        "name": "Cloud Controls Matrix (CCM)",
        "publisher": "Cloud Security Alliance (CSA)",
        "version": "Ver 3.0.1",
        "releaseDate": "2016-10-05T00:00:00.000+0000",
        "lastModified": "2018-05-28T11:20:10.000+0000"
    },
    "requirements": [
        {
            "document": {
                "id": 5443,
                "complianceDocumentId": 2481,
                "section": "AIS",
                "comments": "Application & Interface Security"
            }
        }
    ]
}
```
CloudView APIs

"lastModified": "2016-12-21T15:22:45.000+0000"
},

"summary": {

"mandatesCount": 1,
"requirementsCount": 16,
"mandateName": "Cloud Controls Matrix (CCM)",
"mandatePassPercent": 15.07,
"accounts": [

{

"name": "GCP123",
"id": "100d7969-371b-308f-a08d-b3442170e378",
"accountId": "my-project-1513669048551",
"cloudType": "GCP"
},

{

"name": "connector 2",
"id": "3d96ec1d-1624-3cc1-abc6-dc41d898e532",
"accountId": "gcp-qualys-demo",
"cloudType": "GCP"
}
],
"controlsCount": 8,
"totalEvaluationsCount": 292,
"policiesCount": 1,
"groups": [

{

"groupUuid": "62edaf7d-4530-324c-be52-372a7acee33e",
"groupName": "sample-azure-grp-1"
}
],
"cloudType": "GCP"}
Get List of All Supported Mandates

/rest/v1/reports/mandates

[GET]

We support 24 mandates and you can fetch the list of all the supported mandates.

Sample - Get the list of all supported mandates

API request

curl -k -X GET -u <username>:<password>
'https://<QualysURL>/cloudview-api/rest/v1/reports/mandates'

Response

[  
  {  
    "id": 4282,  
    "name": "Cybersecurity Maturity Model Certification (CMMC) - Maturity Level 2 (ML2)"
  },  
  {  
    "id": 4283,  
    "name": "Cybersecurity Maturity Model Certification (CMMC) - Maturity Level 3 (ML3)"
  },  
  {  
    "id": 4284,  
    "name": "Cybersecurity Maturity Model Certification (CMMC) - Maturity Level 4 (ML4)"
  },  
  {  
    "id": 4285,  
    "name": "Cybersecurity Maturity Model Certification (CMMC) - Maturity Level 5 (ML5)"
  },  
  {  
    "id": 2441,  
    "name": "Payment Card Industry Data Security Standard (PCI-DSS)"
  },]
CloudView APIs

```json
{
    "id": 3561,
    "name": "NIST Cyber Security Framework (CSF)"
},
{
    "id": 3482,
    "name": "Criminal Justice Information Services (CJIS) Security Policy"
},
{
    "id": 2481,
    "name": "Cloud Controls Matrix (CCM)"
},
{
    "id": 3481,
    "name": "IRS Publication 1075"
},
{
    "id": 3381,
    "name": "Minimum Acceptable Risk Standards for Exchanges (MARS-E)"
},
{
    "id": 4061,
    "name": "Monetary Authority of Singapore (MAS) - Notice 834: Cyber Hygiene Practices"
},
{
    "id": 4281,
    "name": "Cybersecurity Maturity Model Certification (CMMC) - Maturity Level 1 (ML1)"
},
{
    "id": 2801,
    "name": "APRA Prudential Practice Guide (PPG): CPG 234 - Management of Security Risk in Information and Information Technology"
},
{
    "id": 2701,
},
{
    "id": 2722,
    "name": "The Australian Signals Directorate - The Essential 8 Strategies (ASD 8)"
}
```
[{"id": 2501, "name": "NERC Critical Infrastructure Protection (CIP)" },
{ "id": 2721, "name": "ANSSI 40 Essential Measures for a Healthy Network"
},
{ "id": 3501, "name": "SWIFT Customer Security Controls Framework - Customer Security Programme"
},
{ "id": 2741, "name": "Reserve Bank of India (RBI) - Baseline Cyber Security and Resilience Requirements (Annex 1)"
},
{ "id": 3401, "name": "NCSC Basic Cyber Security Controls (BCSC)"
},
{ "id": 2443, "name": "ISO/IEC 27001:2013"
},
{ "id": 2762, "name": "NESA UAE Information Assurance Standards (IAS)"
},
{ "id": 2761, "name": "NIST 800-171 (Special Publication)"
},
{ "id": 3861, "name": "Sarbanes-Oxley Act: IT Security"
},
{ "id": 4302, "name": "CIS Controls"
},
{ "id": 2803, "name": "General Data Protection Regulation (GDPR)"
}]
{  
  "id": 2605,
  "name": "Federal Risk and Authorization Management Program (FedRAMP M) - Moderate Security Baseline"
},
{  
  "id": 2604,
  "name": "Federal Risk and Authorization Management Program (FedRAMP H) - High Security Baseline"
},
{  
  "id": 2802,
  "name": "IRDAI Guidelines On Information and Cyber Security for Insurers"
}
]
Response Code: 200
Get List of All Supported Policies

/rest/v1/reports/policies

[GET]

You can fetch the list of all the supported policies in your account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudType</td>
<td>Specify the cloud provider for which you want to fetch all the supported policies. You could mention AWS, AZURE, or GCP.</td>
</tr>
</tbody>
</table>

Sample - Get the list of all the supported policies for Azure

API request

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/reports/policies?cloudType=AZURE'

Response

[
  {
    "id": "df3597f0-9e29-11e9-bdf0-23c5141152bc",
    "title": "CIS Microsoft Azure Foundations Benchmark",
    "cloudType": "AZURE"
  },
  {
    "id": "f9e43730-aedd-11ea-a4bd-5b1aa4c88a83",
    "title": "Azure Database Service Best Practices Policy",
    "cloudType": "AZURE"
  },
  {
    "id": "44441240-3b6c-11ea-93a5-4d1356013529",
    "title": "Azure Function App Best Practices Policy",
    "cloudType": "AZURE"
  }
]
CloudView APIs

[{
  "id": "e6d7e9d0-d476-11e9-bdc5-2f0d02006ccb",
  "title": "Azure Best Practices Policy",
  "cloudType": "AZURE"
},
{
  "id": "e48922a0-ca6d-11ea-992e-a7c52eacc973",
  "title": "Policy",
  "cloudType": "AZURE"
}]

Response Code: 200
Get Report Configurations

/rest/v1/reports

[GET]

You can fetch all the list of report configurations.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Sample - Get the list of report configurations

**API request**

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/reports?pageNo=0&pageSize=3'

**Response**

```json
{
   "content": [
     {
       "reportId": "84685b50-4d43-11e9-9496-d5e5ac80c6e2",
       "title": "Mandate_Report",
       "type": "MANDATE",
       "format": "ON_SCREEN",
       "accounts": [
         {
           "name": "User_John",
           "id": "5a4f0630-39ab-11e9-a7c7-6f7103922bbf",
           "accountId": "111111111111",
           "cloudType": "AWS"
         }
     ]
   }
}```
... "pageable": {
    "sort": {
        "sorted": false,
        "unsorted": true
    },
    "pageSize": 6,
    "pageNumber": 0,
    "offset": 0,
    "paged": true,
    "unpaged": false
},
    "last": false,
    "totalPages": 2,
    "totalElements": 11,
    "first": true,
    "sort": {
        "sorted": false,
        "unsorted": true
    },
    "numberOfElements": 6,
    "size": 6,
    "number": 0
}"
Get Report Details

/rest/v1/reports/{reportId}

[GET]

Specify the report ID and then you can get the details of the specified report.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportId</td>
<td>Unique identifier associated with every report.</td>
</tr>
</tbody>
</table>

Sample - Get the details of specified report configuration

**API request**

curl \-k \-X GET \-u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/reports/01164660-cfc7-11ea-a573-4395559d998e'

**Response**

```json
{
    "mandate": {
        "id": 2481,
        "name": "Cloud Controls Matrix (CCM)",
        "publisher": "Cloud Security Alliance (CSA)",
        "version": "Ver 3.0.1",
        "releaseDate": "2016-10-05T00:00:00.000+0000",
        "lastModified": "2018-05-28T11:20:10.000+0000"
    },
    "requirements": [
        {
            "document": {
                "id": 5443,
                "complianceDocumentId": 2481,
                "section": "AIS",
                "comments": "Application & Interface Security",
                "lastModified": "2016-12-21T15:22:45.000+0000"
            }
        }
    ]
}
```
"summary": {
  "mandatesCount": 1,
  "requirementsCount": 16,
  "mandateName": "Cloud Controls Matrix (CCM)",
  "mandatePassPercent": 15.07,
  "accounts": [
    {
      "name": "GCP123",
      "id": "100d7969-371b-308f-a08d-b3442170e378",
      "accountId": "my-project-1513669048551",
      "cloudType": "GCP"
    },
    {
      "name": "connector 2",
      "id": "3d96ec1d-1624-3cc1-abc6-dc41d898e532",
      "accountId": "gcp-qualys-demo",
      "cloudType": "GCP"
    }
  ],
  "controlsCount": 8,
  "totalEvaluationsCount": 292,
  "policiesCount": 1,
  "groups": [
    {
      "groupUuid": "62edaf7d-4530-324c-be52-372a7acee33e",
      "groupName": "chloe-azure-grp-1"
    }
  ],
  "cloudType": "GCP"
}
Create a Report

/rest/v1/reports

[POST]

To generate a new report you need to provide information such as the cloud provider for which you would want to generate the report and few other details such as name, description, format, mandate ID and so on.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configurationBody</td>
<td>You need to provide the required details in the configurationBody parameter. The syntax for the same is given below:</td>
</tr>
</tbody>
</table>

Syntax:

```json
{
    "cloudType": "string",
    "connectorIds": [
        "string"
    ],
    "description": "string",
    "format": "string",
    "groupIds": [
        "string"
    ],
    "mandateId": "string",
    "policies": [
        {
            "cloudType": "string",
            "policyId": "string"
        }
    ],
    "title": "string",
    "type": "string"
}
```

where,
**cloudType**: the cloud provider (AWS, Azure, or GCP)

**connectorIds**: connector Id

**description**: description of the report

**format**: the report format (only On-Screen format supported)

**groupId**: unique Id of the (connector) group

**mandateId**: unique Id associated with the mandate.

**cloudType**: the cloud provider (AWS, Azure, or GCP)

**policyId**: unique ID associated with the policy.

**title**: name of the report

**type**: indicates if it is policy report or mandate report.

Sample - Create a new report

**API request**

```bash
curl -X POST --header
   -d '{
      "cloudType": "GCP",
      "connectorIds": [
        "100d7969-371b-308f-a08d-b3442170e378","3d96ec1d-1624-3cc1-abc6-
        dc41d898e532"
      ],
      "description": "sample_description",
      "format": "ON_SCREEN",
      "groupId": ["62edaf7d-4530-324c-be52-372a7acee33e"
      ],
      "mandateId": "2481",
      "policies": [
        {
          "cloudType": "GCP",
          "policyId": "10fffb910-3b6d-11ea-93a5-4d1356013529"
        }
      ]
    }
'"}
```
Response

```json
{
  "title": "gcppublicapi",
  "type": "MANDATE",
  "format": "ON_SCREEN",
  "accounts": [
    {
      "name": "GCP123",
      "id": "100d7969-371b-308f-a08d-b3442170e378",
      "accountId": "my-project-XXXXXXXXXXXXX",
      "cloudType": "GCP"
    },
    {
      "name": "connector 2",
      "id": "3d96ec1d-1624-3cc1-abc6-dc41d898e532",
      "accountId": "Demo_GCP",
      "cloudType": "GCP"
    }
  ],
  "description": "string",
  "policies": [
    {
      "policyId": "10ffb910-3b6d-11ea-93a5-4d1356013529",
      "cloudType": "GCP"
    }
  ],
  "mandateId": "2481",
  "createdOn": "2020-07-27T05:07:01+0000",
  "cloudType": "GCP",
  "groupIds": [
    "62edaf7d-4530-324c-be52-372a7acee33e"
  ],
  "reportId": "01164660-cfc7-11ea-a573-4395559d998e"
}
```
Update a Report

/rest/v1/reports/{reportId}

[PATCH]

You can update a report template to generate a new report.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportId</td>
<td>Unique identifier associated with every report.</td>
</tr>
<tr>
<td>configurationBody</td>
<td>You need to provide the required details in the configurationBody parameter.</td>
</tr>
</tbody>
</table>

You need to provide the required details in the configurationBody parameter. The syntax for the same is given below:

Syntax:

```json
{
  "cloudType": "string",
  "connectorIds": [
    "string"
  ],
  "description": "string",
  "format": "string",
  "groupIds": [
    "string"
  ],
  "mandateId": "string",
  "policies": [
    {
      "cloudType": "string",
      "policyId": "string"
    }
  ],
  "title": "string",
  "type": "string"
}
```

where,
**cloudType**: the cloud provider (AWS, Azure, or GCP)

**connectorIds**: connector Id

**description**: description of the report

**format**: the report format (only On-Screen format supported)

**groupId**: unique Id of the (connector) group

**mandateId**: unique Id associated with the mandate.

**cloudType**: the cloud provider (AWS, Azure, or GCP)

**policyId**: unique ID associated with the policy.

**title**: name of the report

**type**: indicates if it is policy report or mandate report.

Sample - Update the configuration of an existing report

```
API request

curl -X PATCH -u <username>:<password> 'Content-Type: application/json'-d '
{
   "cloudType": "GCP",
   "connectorIds": [
      "100d7969-371b-308f-a08d-b3442170e378"
   ],
   "description": "Update Report",
   "format": "ON_SCREEN",
   "groupId": [
      "62edaf7d-4530-324c-be52-372a7acee33e"
   ],
   "mandateId": "2481",
   "policyId": [
      {"cloudType": "GCP",
       "policyId": "10ffb910-3b6d-11ea-93a5-4d1356013529"
      }
   ]
}''
```
CloudView APIs

Response

```json
{
  "title": "gcppublicapi",
  "type": "MANDATE",
  "format": "ON_SCREEN",
  "accounts": [
    {
      "name": "GCP123",
      "id": "100d7969-371b-308f-a08d-b3442170e378",
      "accountId": "my-project-1513669048551",
      "cloudType": "GCP"
    }
  ],
  "description": "string",
  "policies": [
    {
      "policyId": "10ff9b10-3b6d-11ea-93a5-4d1356013529",
      "cloudType": "GCP"
    }
  ],
  "mandateId": "2481",
  "createdAt": "2020-07-24T13:54:19+0000",
  "cloudType": "GCP",
  "groupIds": [
    "62edaf7d-4530-324c-be52-372a7acee33e"
  ],
  "reportId": "24c6d800-cdb4-11ea-8fa8-0f89bf5c84f3"
}
```
Delete Reports

/rest/v1/reports

[DELETE]

Specify the ID of the report you want to delete and the report gets deleted.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportId</td>
<td>Unique identifier associated with every report.</td>
</tr>
</tbody>
</table>

Sample - Delete the specified report

**API request**

```
curl -X DELETE -u <username>:<password> -d '["9cce6540-4b36-11e9-be40-09d60abc9fcd"]' 'https://<QualysURL>/cloudview-api/rest/v1/reports'
```

**Response**

No Content
Response Code: 204

Sample - Delete multiple reports

**API request**

```
curl -X DELETE -u <username>:<password> -d '["9cce6540-4b36-11e9-be40-09d60abc9fcd","fbfd2de0-4af4-11e9-9fd1-1344989d5139"]' 'https://<QualysURL>/cloudview-api/rest/v1/reports'
```

**Response**

No Content
Response Code: 204
# Resource Inventory

**AWS Resources**

/rest/v1/resource/{resourceType}/AWS

[GET]

Fetch all the resources belonging to the specified type in your cloud environment and list the same in the response.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the resources by providing a query using Qualys syntax. The following search token is supported. <a href="#">List of tokens supported for AWS resources</a></td>
</tr>
<tr>
<td>pageNo</td>
<td>(mandatory) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(mandatory) (integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>sort</td>
<td>(keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc.</td>
</tr>
<tr>
<td>updated</td>
<td>Use a date range or specific date to define when the resource was last updated. Examples: Show resources updated within certain dates, updated:</td>
</tr>
</tbody>
</table>
CloudView APIs

[2021-01-01 ... 2021-03-01]

Show resources updated starting 2018-10-01, ending 1 month ago, updated: [2021-01-01 ... now-1m]

Show resources updated starting 2 weeks ago, ending 1 second ago, updated: [now-2w ... now-1s]

Show resources updated on specific date, updated: 2021-01-08

resourceType  (mandatory) Select the type of resource you want to fetch inventory on.

AWS Resource Types: AUTO_SCALING_GROUP, BUCKET, EBS, EC2_INSTANCE, EKS_CLUSTER, EKS_FARGATE_PROFILE, EKS_NODEGROUP, IAM_USER, INTERNET_GATEWAY, LAMBDA, LOAD_BALANCER, NETWORK_ACL, RDS, ROUTE_TABLE, SUBNET, VPC, VPC_SECURITY_GROUP

Sample - Get the list of all AWS resources by type

API request

curl -X GET -u <username>:<password> "https://<QualysBaseURL>/cloudviewapi/rest/v1/resource/BUCKET/AWS?pageNo=0&pageSize=50"

JSON Response

{  "content": [    {      "accountAlias": "sample-XXXXXXXXXXXX",      "controlsFailed": 6,      "bucketName": "aws-cloudtrail-events",      "resourceId": "aws-cloudtrail-events",      "connectorUuids": [        "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX"
      ],      "bucketCreationDateStr": "2021-02-23T00:34:11+0000",      "created": "2022-01-05T11:28.200+00:00"  }  ]
"cloudAccountId": "XXXXXXXXXXXX",
"s3GrantList": [
  {
    "emailAddress": null,
    "groupUri": null,
    "displayName": null,
    "permission": "FullControl",
    "id": "dcf00289423844232d18c426dc98979a5d581505165de60971d8e1a891a44ef7"
  }
],
"uuid": "a5ad9d67-4d1f-3ee2-b625-f3b04a237a8f",
"connectorUuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXX",
"createdOn": "2022-01-05T11:20:28+0000",
"tags": [],
"remediationEnabled": null,
"lastUpdated": "2022-01-05T11:20:28+0000",
"ownerName": "user_john",
"cloudType": "AWS",
"name": "aws-cloudtrail-events",
"bucketPolicy": ...
"resourceType": "BUCKET"
},
"pageable": {
  "sort": {
    "sorted": false,
    "unsorted": true,
    "empty": true
  },
  "pageNumber": 0,
  "pageSize": 6,
  "offset": 0,
  "paged": true,
  "unpaged": false
},
"last": true,
"totalElements": 6,
"totalPages": 1,
"sort": {
  "sorted": false,
  "unsorted": true,
  "empty": true
},
"numberOfElements": 6,
"first": true,
"size": 6,
"number": 0,
"empty": false
}
Azure Resources

/rest/v1/resource/{resourceType}/Azure

[GET]

Fetch all the resources belonging to the specified type in your cloud environment and list the same in the response.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the resources by providing a query using Qualys syntax. The following search token is supported. List of tokens supported for Microsoft Azure Resources</td>
</tr>
<tr>
<td>pageNo</td>
<td>(mandatory) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(mandatory) (integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>sort</td>
<td>(keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc.</td>
</tr>
<tr>
<td>updated</td>
<td>Use a date range or specific date to define when the resource was last updated. Examples: Show resources updated within certain dates, updated: [2021-01-01 ... 2021-03-01]</td>
</tr>
</tbody>
</table>
Show resources updated starting 2018-10-01, ending 1 month ago, updated: [2021-01-01 ... now-1m]

Show resources updated starting 2 weeks ago, ending 1 second ago, updated: [now-2w ... now-1s]

Show resources updated on specific date, updated: 2021-01-08

resourceType  (mandatory) Select the type of resource you want to fetch inventory on.

Azure Resource Types: FUNCTION_APP, NETWORK_SECURITY_GROUP, RESOURCE_GROUP, SQL_SERVER, SQL_SERVER_DATABASE, VIRTUAL_MACHINE, VIRTUAL_NETWORK, WEB_APP

Sample - Get the list of all Azure resources by type

API request

curl -X GET -u <username>:<password> "https://<QualysBaseURL>/cloudviewapi/rest/v1/resource/VIRTUAL_MACHINE/Azure?filter=subscriptionName%3A%22samplesubscription%22&pageNo=0&pageSize=50"

JSON Response

{  "content": [  {   "controlsFailed": 2,   "resourceId": "7de97440-93c5-4ced-9ef9-a3258d2c27da",   "imageData": [  {   "offer": "0001-com-ubuntu-server-focal",   "publisher": "canonical",   "id": null,   "sku": "20_04-lts-gen2",   "version": "latest"  }  ],   "type": "Microsoft.Compute/virtualMachines"  }
"uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX",
"remediationEnabled": null,
"licenseType": null,
"computerName": "Sample-vm-6",
"cloudType": "AZURE",
"customerId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX",
"osType": "Linux",
"customers": [
  "b28e6859-9a15-fb81-833b-d20e458f7f7f"
],
"networkSecurityGroupId": "Sample-vm-6-nsg",
"connectorUuids": [
  "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX"
],
...
"pageable": {
  "sort": {
    "sorted": false,
    "unsorted": true,
    "empty": true
  },
  "pageNumber": 0,
  "pageSize": 5,
  "offset": 0,
  "paged": true,
  "unpaged": false
},
"last": true,
"totalElements": 5,
"totalPages": 1,
"sort": {
  "sorted": false,
  "unsorted": true,
  "empty": true
},
"numberOfElements": 5,
"first": true,
"size": 5,
"number": 0,
"empty": false
}
GCP Resources

/rest/v1/resource/{resourceType}/GCP

[GET]

Fetch all the resources belonging to the specified type in your cloud environment and list the same in the response.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the resources by providing a query using Qualys syntax. The following search token is supported.</td>
</tr>
<tr>
<td>pageNo</td>
<td>(mandatory) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(mandatory) (integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>sort</td>
<td>(keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc.</td>
</tr>
<tr>
<td>updated</td>
<td>Use a date range or specific date to define when the resource was last updated. Examples: Show resources updated within certain dates, updated: [2021-01-01 ... 2021-03-01]</td>
</tr>
</tbody>
</table>
Show resources updated starting 2018-10-01, ending 1 month ago, updated: [2021-01-01 ... now-1m]

Show resources updated starting 2 weeks ago, ending 1 second ago, updated: [now-2w ... now-1s]

Show resources updated on specific date, updated: 2021-01-08

resourceType  (mandatory) Select the type of resource you want to fetch inventory on.

GCP Resource Types: CLOUD_FUNCTION, FIREWALL_RULES, NETWORK, SUBNETWORK, VM_INSTANCE

Sample - Get the list of all GCP resources by type

**API request**


**JSON Response**

```
"content": [ {
    "controlsFailed": 2,
    "resourceId": "2049122088315831723",
    "imageData": null,
    "description": null,
    "type": null,
    "uuid": "41a43830-4061-35f6-9f97-68aa786f9552",
    "zone": "us-central1-a",
    "cloudType": "GCP",
    "customerId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "customers": [ "b28e6859-9a15-fb81-833b-d20e458f7f7f"
                  ],
    "machineType": "e2-micro",
    "connectorUuids": [ "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
                       ]
}
```

"kind": "compute#instance",
"created": "2022-01-05T11:30:07+0000",
"connectorUuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX",
"privateIpAddress": "10.128.15.200",
"tags": null,
"labels": [
{
   "name": "department",
   "value": "engineering"
}
"pageable": {
   "sort": {
      "sorted": false,
      "unsorted": true,
      "empty": true
   },
   "pageNumber": 0,
   "pageSize": 100,
   "offset": 0,
   "paged": true,
   "unpaged": false
},
"last": false,
"totalElements": 195,
"totalPages": 2,
"sort": {
   "sorted": false,
   "unsorted": true,
   "empty": true
},
"numberOfElements": 100,
"first": true,
"size": 100,
"number": 0,
"empty": false
}
Resource Details for AWS by Resource UUID

rest/v1/resource/{resourceType}/uuid/{resourceUuid}/AWS

[GET]

Fetch resource details for AWS resources using resource UUID. To know UUID of a resource, use Get Resource Inventory for AWS API.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the resources by providing a query using Qualys syntax. The following search token is supported.</td>
</tr>
<tr>
<td>pageNo</td>
<td>(mandatory) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(mandatory) (integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>sort</td>
<td>(keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc.</td>
</tr>
</tbody>
</table>
| updated    | Use a date range or specific date to define when the resource was last updated.  
Examples:  
Show resources updated within certain dates, updated: [2021-01-01 ... 2021-03-01]  
Show resources updated starting 2018-10-01, ending 1 month ago, updated: [2021-01-01 ... now-1m] |
Show resources updated starting 2 weeks ago, ending 1 second ago, updated: [now-2w ... now-1s]

Show resources updated on specific date, updated: 2021-01-08

resourceType (mandatory) Select the type of resource you want to fetch inventory on.

AWS Resource Types: AUTO_SCALING_GROUP, BUCKET, EBS, EC2_INSTANCE, EKS_CLUSTER, EKS_FARGATE_PROFILE, EKS_NODEGROUP, IAM_USER, INTERNET_GATEWAY, LAMBDA, LOAD_BALANCER, NETWORK_ACL, RDS, ROUTE_TABLE, SUBNET, VPC, VPC_SECURITY_GROUP

resourceUUID (mandatory only when you fetch resource details). Specify the unique resource ID.

You can fetch the resource UUID by using the Get Resource Inventory API
(/cloudviewapi/rest/v1/resource/{resourceType}/{cloudprovider})

Sample - Get resource details for resource of type S3 bucket by resource UUID

API request

curl -X GET -u <username>:<password>
"https://<QualysBaseURL>/cloudview-api/rest/v1/resource/BUCKET/uuid/96ca11d8-1c26-365d-b644-355ae2b8b588/AWS"

JSON Response

```json
{
  "uuid": "96ca11d8-1c26-365d-b644-355ae2b8b588",
  "connectorUuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "connectorUuuids": [
    "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
  ],
  "region": "eu-south-1",
  "name": "aws-cloudtrail-logs-raghav-trail-events",
  "cloudType": "AWS",
  "createdOn": "2022-01-05T11:20:28+0000",
}
"created": "2022-01-05T11:20:28.200+00:00",
"remediationEnabled": null,
"lastUpdated": "2022-01-05T11:20:28+0000",
"cloudAccountId": "XXXXXXXXXXXX",
"accountAlias": "sample_alias",
"tags": [],
"resourceId": "aws-cloudtrail-events",
"controlsFailed": 6,
"bucketName": "aws-cloudtrail-events",
"ownerName": "user_john",
"bucketOwnerId": "dcf00289423844232d18c426dc98979a5d581505165de60971d8e1a891a44ef7",
"bucketCreationDateStr": "2021-02-23T00:34:11+0000",
"s3GrantList": [
{
"displayName": null,
"id": "dcf00289423844232d18c426dc98979a5d581505165de60971d8e1a891a44ef7",
"emailAddress": null,
"groupUri": null,
"permission": "FullControl"
}
],
"resourceType": "BUCKET"
Resource Details for Azure by Resource UUID

/rest/v1/resource/{resourceType}/uuid/{resourceUuid}/Azure

[GET]

Fetch resource details for Azure resources using resource UUID. To know UUID of a resource, use Get Resource Inventory for Azure API.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the resources by providing a query using Qualys syntax. The following search token is supported. <a href="#">List of tokens supported for Microsoft Azure Resources</a></td>
</tr>
<tr>
<td>pageNo</td>
<td>(mandatory) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(mandatory) (integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>sort</td>
<td>(keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc.</td>
</tr>
<tr>
<td>updated</td>
<td>Use a date range or specific date to define when the resource was last updated. Examples: Show resources updated within certain dates, updated: [2021-01-01 ... 2021-03-01] Show resources updated starting 2018-10-01, ending 1 month ago, updated: [2021-01-01 ... now-1m]</td>
</tr>
</tbody>
</table>
CloudView APIs

Show resources updated starting 2 weeks ago, ending 1 second ago, updated: [now-2w ... now-1s]

Show resources updated on specific date, updated: 2021-01-08

resourceType  (mandatory) Select the type of resource you want to fetch inventory on.

Azure Resource Types: FUNCTION_APP, NETWORK_SECURITY_GROUP, RESOURCE_GROUP, SQL_SERVER, SQL_SERVER_DATABASE, VIRTUAL_MACHINE, VIRTUAL_NETWORK, WEB_APP

resource UUID  (mandatory only when you fetch resource details). Specify the unique resource ID.

You can fetch the resource UUID by using the Get Resource Inventory API (/cloudviewapi/rest/v1/resource/{resourceType}/<cloudprovider>)

Sample - Get resource details for resource of type Virtual Machine by resource UUID

API request

```
curl -X GET -u <username>:<password> "https://<QualysBaseURL>/cloudview-api/rest/v1/resource/VIRTUAL_MACHINE/uuid/8513b79d-8a53-3642-a2ba-aca31886eeaf/Azure"
```

JSON Response

```
{
    "uuid": "8513b79d-8a53-3642-a2ba-aca31886eeaf",
    "connectorUuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "connectorUuids": [
        "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
    ],
    "cloudType": "AZURE",
    "customerId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "customers": [
        "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
    ],
    "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
}```
"subscriptionName": "sample_azure",
"resourceId": "7de97440-93c5-4ced-9ef9-a3258d2c27da",
"resourceGroupName": "CloudView_QA",
"scanUuid": "58d07687-4213-43ff-a4c1-f39c8f214943",
"name": "Sample-vm-6",
"type": "Microsoft.Compute/virtualMachines",
"region": "eastus",
"tags": [],
"remediationEnabled": null,
"controlsFailed": 2,
"primaryPublicIPAddress": "20.124.231.2",
"primaryPublicIPAddressId": "/subscriptions/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX/resourceGroups/CloudView_QA/providers/Microsoft.Network/publicIPAddresses/sample-vm-6-ip",
"availabilitySetId": "",
"provisioningState": null,
"licenseType": null,
"computerName": "sample-vm-6",
"size": "Standard_B1s",
"osType": "Linux",
"statuses": [
  {
    "code": "ProvisioningState/succeeded",
    "displayStatus": "Provisioning succeeded",
    "level": "INFO",
    "message": null,
    "time": "2022-01-04T05:26:42+0000"
  },
  {
    "code": "PowerState/running",
    "displayStatus": "VM running",
    "level": "INFO",
    "message": null,
    "time": null
  }
],
"created": "2022-01-04T06:20:11+0000",
"updated": "2022-01-05T11:28:17+0000",
"networkSecurityGroupId": "sample-vm-6-nsg",
"imageData": [
  {
    "id": null,
    "offer": "0001-com-ubuntu-server-focal",
    "publisher": "canonical",
    "sku": "20_04-lts-gen2",
    "version": "latest"
CloudView APIs
Resource Details for GCP by Resource UUID

/rest/v1/resource/{resourceType}/uuid/{resourceUuid}/GCP

[GET]

Fetch resource details for GCP using resource UUID. To know UUID of a resource, use Get Resource Inventory for GCP API.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the resources by providing a query using Qualys syntax. The following search token is supported. <a href="#">List of tokens supported for GCP Resources</a></td>
</tr>
<tr>
<td>pageNo</td>
<td>(mandatory) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(mandatory) (integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>sort</td>
<td>(keyword) Sort the results using a Qualys token. Sorting is currently enabled with only one sort token: lastSyncedOn. The allowed values are asc or desc.</td>
</tr>
</tbody>
</table>
| updated   | Use a date range or specific date to define when the resource was last updated. Examples:  

Show resources updated within certain dates, updated: [2021-01-01 ... 2021-03-01]  

Show resources updated starting 2018-10-01, ending 1 month ago, updated: [2021-01-01 ... now-1m]  

Show resources updated starting 2 weeks ago, ending 1 second |
CloudView APIs

resourceType (mandatory) Select the type of resource you want to fetch inventory on.

GCP Resource Types: CLOUD_FUNCTION, FIREWALL_RULES, NETWORK, SUBNETWORK, VM_INSTANCE

resource UUID (mandatory only when you fetch resource details). Specify the unique resource ID.

You can fetch the resource UUID by using the Get Resource Inventory API (/cloudviewapi/rest/v1/resource/{resourceType}/<cloudprovider>)

Sample - Get resource details for resource of type VM instance by resource UUID

API request

curl -X GET -u <username>:<password> "https://<QualysBaseURL>/cloudview-api/rest/v1/resource/VM_INSTANCE/uuid/715c038e-dc4c-3949-8eee-661fcf559116/GCP"

JSON Response

```json
{
    "uuid": "715c038e-dc4c-3949-8eee-661fcf559116",
    "connectorUuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX",
    "connectorUuids": [
        "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX"
    ],
    "cloudType": "GCP",
    "customerId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX",
    "customers": [
        "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX"
    ],
    "projectId": "my-sample-project-XXXXXXXXXXXXX",
    "resourceId": "204912088315831723",
    "scanUuid": "12b1b7bc-dcb8-4cf4-9424-afb1a16d191b",
    "name": "gke-test-1-9-fail-pool-2-e90f48c4-ukkn"
}
```
"type": null,
"region": "us-central1",
"tags": null,
"controlsFailed": 2,
"zone": "us-central1-a",
"kind": "compute#instance",
"machineType": "e2-micro",
"description": null,
"externalIpAddress": "34.132.50.15",
"privateIpAddress": "10.128.15.200",
"labels": [
  {
    "name": "department",
    "value": "engineering"
  },
  {
    "name": "manager",
    "value": "sample_jim"
  },
  {
    "name": "owner",
    "value": "user_john"
  },
  {
    "name": "team",
    "value": "sample_controls"
  },
  {
    "name": "test",
    "value": "test20"
  }
],
"networkInterfaces": [
  {
    "accessConfigs": {
      "kind": null,
      "name": null,
      "type": null,
      "networkTier": null
    },
    "kind": "compute#networkInterface",
    "name": "nic0",
    "network": "vj",
    "networkIP": "10.128.15.200",
    "subnetwork": "vj",
    "fingerprint": "x-W6Anib-M="
  }
]
"status": "RUNNING",
"created": "2022-01-05T11:30:07+0000",
"updated": "2022-01-05T11:30:07+0000",
"imageData": null
Alerting Response APIs

Alerting Response APIs (Beta)

You can configure monitoring of critical controls and triggering alert messages on detection of critical conditions. The alert messages you receive includes control assessment details.

To receive the alerts, you need to follow quick steps:

- Create one or more actions.

- Create rules that include criteria or specific conditions that would trigger the alert and associate actions for each criterion.

- Run the Connectors in CloudView and the alerts get triggered whenever the condition defined in a Rules are satisfied.

Based on action type you select, you will be notified through Email, Slack, or Pagerduty.

Response Actions

Response Notifications

Response Rules
Response Actions

We support the following response actions:

- Get Actions
- Get Action by Id
- Delete Actions
- Create email Action
- Update email Action
- Create PagerDuty Action
- Update PagerDuty Action
- Test PagerDuty Action
- Create Slack Action
- Update Slack Action
- Test Slack Action
- Get all Action Types
Get Actions
/rest/v1/actions/
[GET]

You can get the list of actions using this API. You can search for actions using filters based on criteria you want.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| filter    | Form the search query using the filters we provide to refine the search for actions. Filters supported:  
  action.name  
  action.description  
  action.type  
  action.createdBy  
  action.createdById  
  action.updatedBy  
  action.updatedById  
  action.active  
  action.disabled  
  action.createdDate  
  action.updatedDate  
  action.emailRecipient  
  action.subject  
  action.slackChannel  
  action.slackWebhookUri  
  action.pagerdutyServiceKey  
For detailed information on filters, see the Reference: Action Filters. |
| pageNo    | (integer) The page to be returned. |
| pageSize  | (integer) The number of records per page to be included in the response. |
sortField
Sort by a specific field.

sortOrder
Specify if the sorting needs to be ascending or descending order.

Sample - Get the list of actions

Let us get the actions that are created by a specific user.

API request

curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQK==' 'https://<QualysURL>/cloudview-api/rest/v1/actions?filter=action.createdBy%3Duser_john&pageNo=1&pageSize=50&sortOrder=asc'

Response

```json
{
  "id": "24278970-725c-11ea-9959-f36a27b72f5a",
  "name": "string12345",
  "description": "Sample Pager",
  "actionType": "pagerduty",
  "createdBy": "John Doe",
  "createdById": "user_john",
  "updatedBy": "John Doe",
  "updatedById": "user_john",
  "created": "2020-03-30T07:57:45.735+0000",
  "updated": "2020-03-30T08:07:35.896+0000",
  "alert": "Qualys CloudView: Cloud Security Assessment Alerts

Severity Control Failure Detected for CID ${cid}

*Affected Resource*
  resourceId:${resource.id}
  resourceType:${resource.type}
  service:${service.type}
  region:${region}
  cloudType:${provider.type}
  account:${account.id}
  connector:${connector.id}
  group:${accountGroup}

*Evaluation Summary*
  controlName:${control.name}
  controlId:${control.id}
  policyName:${policyName}
  evaluatedOn:${evaluatedOn}
  firstEvaluated:${firstEvaluated}
  lastEvaluated:${lastEvaluated}
  results:
    result:${control.result}
    evidences:
      settingName:${settingName}
      actualValue:${actualValue}

Yours
```
Sincerely,
Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "Sample Pager Action",
"pagerdutyServiceKey": "c391356a9d7d4c6b8aa0257ff91cc3842",
"pagerdutyEventType": "trigger",
"activeRules": 0,
"disabledRules": 0
},
{
  "id": "36bc5690-6dccc-11ea-97c4-57de4ff3e879",
  "name": "Azure Action",
  "description": "Azure Action",
  "actionType": "qemail",
  "createdBy": "John Doe",
  "createdById": "user_john",
  "updatedBy": "John Doe",
  "updatedById": "user_john",
  "created": "2020-03-24T12:37:24.729+0000",
  "updated": "2020-03-24T12:37:24.729+0000",
  "alert": "Qualys CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "Azure CV Test",
"smtpHost": "mta01.eng.abc01.example.com",
"smtpPort": 25,
"emailRecipients": [ "abc@example.com"
],
"emailFromAddress": "noreply@example.com",
"emailReplyTo": "noreply@example.com",
"activeRules": 0,
"disabledRules": 0
},
{
  "id": "1f695df0-6da2-11ea-8910-77b847f40d61",
  "name": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours Sincerely,
"name": "Sample Slack",
"description": "Sample Slack description",
"actionType": "slack",
"createdBy": "John Doe",
"createdById": "user_john",
"updatedBy": "John Doe",
"updatedById": "user_john",
"created": "2020-03-24T07:36:06.735+0000",
"updated": "2020-03-30T07:54:43.371+0000",
"alert": "Qualys CloudView: Cloud Security Assessment
Alerts

*${control.criticality} Severity Control Failure Detected for CID ${cid}*

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accountId:${account.id}
	connectorId:${connectorUuid}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
	settingName:${evidences.key}
	actualValue: ${evidences.value}

Yours Sincerely,
Qualys Support Team

For any assistance, please contact our <mailto:support@qualys.com | customer support team.>

"slackWebhookUri": "https://hooks.slack.com/services/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",
"slackChannel": "Sample-slack",
"activeRules": 1,
"disabledRules": 0
}
Get Action by Id

/rest/v1/actions/{actionId}

[GET]

View details for a specific action which is in the user’s scope.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionId</td>
<td>(mandatory) (integer) Specify the action ID of an action in the user’s scope.</td>
</tr>
</tbody>
</table>

Sample - Get action details using the action Id

Let us fetch details of a Slack action using the action Id.

API request

```
curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/actions/bd786210-9965-11e8-ab43-6187ace8f6e8'
```

Response

```
{
    "id": "1f695df0-6da2-11ea-8910-77b847f40d61",
    "name": "Sample action",
    "description": "Action details",
    "actionType": "slack",
    "createdBy": "John Doe",
    "createdById": "user_john",
    "updatedBy": "John Doe",
    "updatedById": "user_john",
    "created": "2020-03-24T07:36:06.735+0000",
    "updated": "2020-03-30T07:54:43.371+0000",
    "alert": "Qualys CloudView: Cloud Security Assessment Alerts\n\n$\{control.criticality\} Severity Control Failure Detected for CID $\{cid\}*$\n\n$\{resource.id\}$ Resource*
\n$\{resource.id\}$\n\n$\{resource.type\}$
```
CloudView APIs

```json
{
  "service": "${service.type}",
  "region": "${region}",
  "cloudType": "${provider.type}",
  "accountId": "${account.id}",
  "connectorId": "${connectorUuid}",
  "Evaluation Summary":
    "controlName": "${control.name}",
    "controlId": "${cid}",
    "policyName": "${policyName}",
    "evaluatedOn": "${evaluatedOn}",
    "evaluationDates": {
      "firstEvaluated": "${firstEvaluated}",
      "lastEvaluated": "${lastEvaluated}"}

  *Results*
    "result": "${control.result}",
    "evidences": {
      "settingName": "${evidences.key}",
      "actualValue": "${evidences.value}"}

Yours Sincerely,
Qualys Support Team

For any assistance, please contact our <mailto:support@qualys.com | customer support team.>

"slackWebhookUri": "https://hooks.slack.com/services/T95RLRTSL/BRD8PB06/oxQZYabcBEIex6Mh0R6mMxyz",
"slackChannel": "sample-slack",
"activeRules": 1,
"disabledRules": 0
}
```
Delete Action

/rest/v1/actions/delete

[POST]

Specify the ID of an existing action you want to delete and the action gets deleted. Ensure that action you want to delete is not associated with a rule in use.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionId</td>
<td>(Array [string]) Specify the ID of an action to be deleted and the action gets deleted. You can provide multiple Ids separated by comma.</td>
</tr>
</tbody>
</table>

Example:

```
{
  "ids": [
    "actionId1,
    "actionId2"
  ]
}
```

Sample - Delete a specific action

API request

```
curl -k -X POST-u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/actions/delete'
```

Request POST Data

```
{
  "ids": [
    "bd786210-9965-11e8-ab43-6187ace8f6e8",
    "efbf4080-52dd-11ea-a008-cbe911ab6a51"
  ]
}
```
## Response

No Content
Response Code: 200
Create email Action

/rest/v1/actions/email

[POST]

You can create an alert to be sent through email (action type: qemail). Specify the necessary details in the request body that are required to create an email action such as action name, action description, the recipient details, whom the email should be sent to and so on.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emailActionRequest</td>
<td>(body) Specify the actionName, actionDescription, and recipients and so on. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```
{
    "actionDescription": "string",
    "actionName": "string",
    "recipients": [
        "example@abc.com"
    ],
    "subject": "string"
}
```

Where,

- actionDescription: description that tells the purpose of the action
- actionName: name of the action
- recipients: valid email ID of the recipients to whom the alert should be sent. You can provide multiple email IDs separated by comma.
- subject: subject of the email action

Sample - Create email action
API request

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/actions/email'

Request POST Data

```
{
    "actionDescription": "Sample Action Test",
    "actionName": "Sample action",
    "recipients": [
        "user_john@example.com"
    ],
    "subject": "Sample Alert"
}
```

Response

```
{
    "success": "bd786210-9965-11e8-ab43-6187ace8f6e8"
}
```
Update email Action

/rest/v1/actions/email/{emailActionId}

[POST]

You can update email action. Specify the necessary details in the request body that are required to update an email action such as action ID, action name, action description, the recipient details, whom the email should be sent to and so on.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emailActionId</td>
<td>(mandatory) Specify the ID of the email action that you want to update.</td>
</tr>
<tr>
<td>emailActionRequest</td>
<td>(body) Specify the actionName, actionDescription, and recipients and so on. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```json
{
    "actionDescription": "string",
    "actionName": "string",
    "recipients": [
        "example@abc.com"
    ],
    "subject": "string"
}
```

Where,

- **actionDescription**: description that tells the purpose of the action
- **actionName**: name of the action
- **recipients**: valid email ID of the recipients to whom the alert should be sent. You can provide multiple email IDs separated by comma.
- **subject**: subject of the email action
Sample - Update the email action

Let us update the description of an existing action.

**API request**

```
curl -k -X PUT -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/actions/email/bd786210-9965-11e8-ab43-6187ace8f6e8'
```

**Request POST Data**

```
{
  "actionDescription": "Update Sample Action Test",
  "actionName": "Sample action",
  "recipients": [
    "user_john@example.com"
  ],
  "subject": "Sample Alert"
}
```

**Response**

```
{
  "success": "bd786210-9965-11e8-ab43-6187ace8f6e8"
}
```
Create PagerDuty Action

/rest/v1/actions/pagerduty

[POST]

You can create an alert to be notified through PagerDuty application. Specify the necessary details in the request body that are required for PagerDuty such as action name, action description, client, and servicekey, and so on.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| pagerdutyRequest| (body) Specify the action actionName, actionDescription, and recipients and so on. Refer to the following example for exact syntax. 

```json
{
  "actionDescription": "string",
  "actionName": "string",
  "client": "string",
  "serviceKey": "string",
  "subjectLine": "string"
}
```

Where,

- actionDescription: description that tells the purpose of the action
- actionName: name of the action
- client:
- serviceKey: the service key required to connect to your PagerDuty account.
- subjectLine: subject of the action

Sample - Create PagerDuty Action
### API request

curl -k -X POST -u <username>:<password> 
'https://<QualysURL>/cloudview-api/rest/v1/actions/pagerduty'

### Request POST Data

```json
{
    "actionDescription": "Sample PagerDuty action",
    "actionName": "Pagerduty action",
    "client": "sample",
    "serviceKey": "c391356a9d7d4c6b8a0257ff91cc3842",
    "subjectLine": "Test Pager action"
}
```

### Response

```json
{
    "success": "bd786210-9965-11e8-ab43-6187ace8f6e8"
}
```
Update PagerDuty Action

/rest/v1/actions/pagerduty/{pagerActionId}

[PUT]

You can update the action to be notified through PagerDuty. Specify the necessary details in the request body that are required for PagerDuty such as action name, action description, the recipient details, whom the alert should be sent to and so on.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pagerActionId</td>
<td>(mandatory) (integer) Specify the pagerAction ID of a specific action in the user’s scope.</td>
</tr>
<tr>
<td>pagerdutyRequest</td>
<td>(body) Specify the action actionName, actionDescription, and recipients and so on. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```json
{
   "actionDescription": "string",
   "actionName": "string",
   "client": "string",
   "serviceKey": "string",
   "subjectLine": "string"
}
```

Where,

- actionDescription: description that tells the purpose of the action
- actionName: name of the action
- client:
- serviceKey: the service key required to connect to your PagerDuty account.
- subjectLine: subject of the action
Sample - Update PagerDuty Action

API request
```
curl -k -X PUT -u <username>:<password>
'https://<QualysURL>/cloudview-api/rest/v1/actions/pagerduty/test'
```

Request PUT Data
```
{
    "actionDescription": "Sample PagerDuty action",
    "actionName": "Pagerduty action",
    "client": "string",
    "serviceKey": "c391356a9d7d4c6b8a0257ff91cc3842",
    "subjectLine": "Test Pager action"
}
```

Response
```
{
    "success": "03e5b680-52f6-11ea-a008-cbe911ab6a51"
}
```
Test PagerDuty Action
/rest/v1/actions/pagerduty/test

[POST]

You can execute a test action to check if PagerDuty is reachable or not.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pagerdutyConnectionParam</td>
<td>(body) Specify the servicekey used to be able to connect to PagerDuty.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;serviceKey&quot;: &quot;c391356a9d7d4c6b8a0257ff91cc3123&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

Sample - Update PagerDuty Action

API request
curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/actions/pagerduty/test'

Request POST Data

```
{
   "serviceKey": "c391356a9d7d4c6b8a0257ff91cc3842"
}
```

Response

```
{
   "success": "true"
}
```
Create Slack Action

/rest/v1/actions/slack

[POST]

You can create an alert to be notified through Slack. Specify the necessary details in the request body that are required for Slack such as action name, action description, the recipient details, whom the alert should be sent to and so on.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slackRequest</td>
<td>(body) Specify the action actionName, actionDescription, and recipients and so on. Refer to the following example for exact syntax.</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;actionDescription&quot;: &quot;string&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;actionName&quot;: &quot;string&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;channel&quot;: &quot;string&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;webhookUri&quot;: &quot;string&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>Where,</td>
</tr>
<tr>
<td></td>
<td>actionDescription: description that tells the purpose of the action</td>
</tr>
<tr>
<td></td>
<td>actionName: name of the action</td>
</tr>
<tr>
<td></td>
<td>channel: the channel name of your slack account</td>
</tr>
<tr>
<td></td>
<td>webhookUri: the Webhook URI required to connect to your slack account to post alert messages.</td>
</tr>
</tbody>
</table>

**Sample - Create Slack Action**

**API request**

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/actions/slack'
### Request POST Data

```json
{
    "actionDescription": "Sample slack action description",
    "actionName": "Sample slack action",
    "channel": "Sample-slack"
    "webhookUri": "https://hooks.slack.com/services/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"
}
```

### Response

```json
{
    "success": "bd786210-9965-11e8-ab43-6187ace8f6e8"
}
```
Update Slack Action

/rest/v1/actions/slack/{slackActionId}

[PUT]

You can update the action to be notified through Slack. Specify the necessary details in the request body that are required for Slack such as action name, action description, the recipient details, whom the alert should be sent to and so on.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slackActionId</td>
<td>(mandatory) (integer) Specify the slackAction ID of a specific action in the user’s scope.</td>
</tr>
<tr>
<td>slackRequest</td>
<td>(body) Specify the action actionName, actionDescription, and recipients and so on. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```json
{
    "actionDescription": "string",
    "actionName": "string",
    "channel": "string",
    "webhookUri": "string"
}
```

Where,

actionDescription: description that tells the purpose of the action

actionName: name of the action

channel: the channel name of your slack account

webhookUri: the Webhook URI required to connect to your slack account to post alert messages.

Sample - Update Slack Action
<table>
<thead>
<tr>
<th>API request</th>
</tr>
</thead>
<tbody>
<tr>
<td>curl -k -X PUT -u &lt;username&gt;:&lt;password&gt; 'https://&lt;QualysURL&gt;/cloudview-api/rest/v1/actions/slack/bd786210-9965-11e8-ab43-6187ace8f6e8'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Request PUT Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;actionDescription&quot;: &quot;Sample slack action description&quot;,</td>
</tr>
<tr>
<td>&quot;actionName&quot;: &quot;Sample slack action&quot;,</td>
</tr>
<tr>
<td>&quot;channel&quot;: &quot;Sample-slack&quot;</td>
</tr>
<tr>
<td>&quot;webhookUri&quot;: &quot;<a href="https://hooks.slack.com/services/T95RLRTSL/BRD6PBJ07/oxQZYxmrBEIex3Mh0R5mM">https://hooks.slack.com/services/T95RLRTSL/BRD6PBJ07/oxQZYxmrBEIex3Mh0R5mM</a> mpl&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;success&quot;: &quot;bd786210-9965-11e8-ab43-6187ace8f6e8&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>
Test Slack Action
/rest/v1/actions/slack/test

[POST]

You can execute a test action to check if Slack is reachable or not.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slackConnectionParam</td>
<td>Specify the channel and webhookUri of your Slack account used to create the action.</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "channel": "Sample Channel",
  "webhookUri": "https://hooks.slack.com/services/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"
}
```

Sample - Test Slack Action

**API request**

```bash
curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/actions/slack/test'
```

**Request POST Data**

```json
{
  "channel": "Sample Slack",
  "webhookUri": "https://hooks.slack.com/services/T95RLRTSL/BRD8PBJ06/oxQZxmrBEIex5Mh0R7mMmpl"
}
```

**Response**

```json
{
  "success": "true"
}
```
Get all Action Types

/rest/v1/actions/types

[GET]

Fetch the list of actions type we support: qemail, Slack, and PagerDuty.

**Sample - Get the list of action types**

**API request**

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQK==' 'https://<QualysURL>/cloudview-api/rest/v1/actions/types'
```

**Response**

```
[
  "qemail",
  "slack",
  "pagerduty"
]
```
Reference: Action Filters

You can form the search query using the filters we provide to refine the search for actions.

action.name

Use quotes or backticks within values to find actions with certain name.

Examples

Find actions with name

action.name: Post to Slack Channel

Find actions that contain parts of the name

action.name: "Post to Slack Channel"

Find actions that match exact value

action.name: `Post to Slack Channel`

action.description

Use quotes or backticks within values to find actions with certain description.

Examples

Find actions with description

action.description: creates alert by posting to slack channel

Find actions that contain parts of the description

action.description: "creates alert by posting to slack channel"

Find actions that match exact value

action.description: `creates alert by posting to slack channel`

action.type

Use a text value ###### to find actions with certain type (Email, slack or pagerduty).
CloudView APIs

Example

Find actions of type

action.type: **SLACK**

**action.createdBy**

Use *quotes or backticks* within values to find actions created by a certain user.

Examples

Find actions created by user

action.createdBy: **Joe Smith**

Find actions that contain parts of the user name

action.createdBy: "Joe Smith"

Find actions that match exact value

action.createdBy: `Joe Smith`

**action.createdById**

Use *quotes or backticks* within values to find actions created by a certain user ID.

Examples

Find actions created by user ID

action.createdById: **jsmith**

Find actions that contain parts of the user ID

action.createdById: "jsmith"

Find actions that match exact value

action.createdById: `jsmith`

**action.updatedBy**
Use **quotes or backticks** within values to find actions updated by a certain user.

**Examples**

Find actions updated by user

```json
action.updatedBy: Joe Smith
```

Find actions that contain parts of the user name

```json
action.updatedBy: "Joe Smith"
```

Find actions that match exact value

```json
action.updatedBy: 'Joe Smith'
```

**action.updatedById**

Use **quotes or backticks** within values to find actions updated by a certain user ID.

**Examples**

Find actions updated by user ID

```json
action.updatedById: jsmith
```

Find actions that contain parts of the user ID

```json
action.updatedById: "jsmith"
```

Find actions that match exact value

```json
action.updatedById: 'jsmith'
```

**action.active**

Use an **Integer value ######** to find actions with certain number of active rules.

**Examples**

Find action with 3 active rules

```json
action.active : 3
```
Find action with more than 3 active rules

```java
action.active > 3
```

**action.disabled**

Use an **integer value #####** to find actions with certain number of disabled rules.

**Examples**

Find action with 3 disabled rules

```java
action.disabled : 3
```

Find action with more than 3 disabled rules

```java
action.disabled > 3
```

**action.createdDate**

Use a **date range** or specific date to find when actions were created.

**Examples**

Show actions created within certain dates

```java
action.createdDate: [2018-02-01 ... 2018-02-12]
```

Show actions created starting 2018-02-01, ending 1 month ago

```java
action.createdDate: [2018-02-01 ... now-1M]
```

Show actions created starting 2 weeks ago, ending 1 second ago

```java
action.createdDate: [now-2w ... now-1s]
```

Show actions created on certain date

```java
action.createdDate:'2018-02-22'
```

**action.updatedDate**

Use a **date range** or specific date to find when actions were last modified.

**Examples**
Show actions updated within certain dates

\texttt{action.updatedDate: } [2018-02-01 ... 2018-02-12]

Show actions updated starting 2018-02-01, ending 1 month ago

\texttt{action.updatedDate: } [2018-02-01 ... now\text{-}1M]

Show actions updated starting 2 weeks ago, ending 1 second ago

\texttt{action.updatedDate: } [now\text{-}2w ... now\text{-}1s]

Show actions updated on certain date

\texttt{action.updatedDate: } '2018-02-22'

\texttt{action.emailRecipient}

Use \textbf{quotes or backticks} within values to find actions with certain email recipients.

\textit{Examples}

Find actions with email recipient

\texttt{action.emailRecipient: secops\text{-}alert@mycompany.com}

Find actions that contain parts of the email recipient

\texttt{action.emailRecipient: "secops\text{-}alert@mycompany.com"}

Find actions that match exact value

\texttt{action.emailRecipient: `secops\text{-}alert@mycompany.com`}

\texttt{action.subject}

Use \textbf{quotes or backticks} within values to find actions with certain text in the subject (email or pagerduty subject).

\textit{Examples}

Find actions with subject

\texttt{action.subject: warning}
Find actions that contain parts of the subject

```javascript
action.subject: "warning"
```

Find actions that match exact value

```javascript
action.subject: 'warning'
```

**action.slackChannel**

Use *quotes or backticks* within values to find actions with certain slack channel name.

*Examples*

Find actions with slack channel

```javascript
action.slackChannel: Sec Ops
```

Find actions that contain parts of the slack channel name

```javascript
action.slackChannel: "Sec Ops"
```

Find actions that match exact value

```javascript
action.slackChannel: 'Sec Ops'
```

**action.slackWebhookUri**

Use *quotes or backticks* within values to find actions with certain Slack Webhook URI.

*Examples*

Find actions with Slack Webhook URI

```javascript
action.slackWebhookUri: https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXX
```

Find actions that contain parts of the Slack Webhook URI

```javascript
action.slackWebhookUri: "https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXX"
```

Find actions that match exact value
action.slackWebhookUri:
`https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXXXX
X`

action.pagerdutyServiceKey

Use **quotes or backticks** within values to find actions with certain pagerduty service key.

*Examples*

Find actions with pagerduty service key

action.pagerdutyServiceKey: 78c52868deb562fcbad765275da

Find actions that contain parts of the pagerduty service key

action.pagerdutyServiceKey: "78c52868deb562fcbad765275da"

Find actions that match exact value

action.pagerdutyServiceKey: `78c52868deb562fcbad765275da`
Response Notifications

We support following actions for the Response Notifications API:

Get Activities

Get Activities by Id
Get Activities

/rest/v1/activities

[GET]

You can get the list of activities using this API. You can view the activities for a particular cloud provider.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloudType</td>
<td>(mandatory) Select the cloud provider from AWS, Azure, or GCP.</td>
</tr>
<tr>
<td>filter</td>
<td>Form the search query using the filters we provide to refine the search for actions.</td>
</tr>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Filters supported:

- ruleName
- rule.description
- status
- statusDate
- aggregate
- createdBy
- createdById
- action.name
- action.type
- action.message
- action.subject
- action.emailRecipient
- action.slackChannel

For detailed information on filters, see the Reference: Action Filters.
sortField | Specify the field that decides the sort order for the actions.

sortOrder | {asc|desc} Specify if the sorting needs to be ascending or descending order.

**Sample - Get the list of actions**

Let us get the actions that are created by a specific user.

**API request**

```bash
curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQK==' 'https://<QualysURL>/cloudview-api/rest/v1/actions?filter=action.createdBy%3Duser_john&pageNo=1&pageSize=50&sortOrder=asc'
```

**Response**

```
{
  "id": "24278970-725c-11ea-9959-f36a27b72f5a",
  "name": "string12345",
  "description": "Sample Pager",
  "actionType": "pagerduty",
  "createdBy": "John Doe",
  "createdById": "user_john",
  "updatedBy": "John Doe",
  "updatedById": "user_john",
  "created": "2020-03-30T07:57:45.735+0000",
  "updated": "2020-03-30T08:07:35.896+0000",
  "alert": "Qualys CloudView: Cloud Security Assessment Alerts

$ {control.criticality} Severity Control Failure Detected for CID ${cid}n\n*n*Affected

\nResource*\n  \n  ${resource.id}\n  ${resource.type}\n  ${service.type}\n  ${region}\n  ${cloudType}\n  ${account.id}\n  ${connectorId}\n  ${groupName}\n*Evaluation

Summary*\n  \n  ${control.name}\n  ${policyName}\n  ${evaluatedOn}\n  ${evaluationDates}\n  ${firstEvaluated}\n  ${lastEvaluated}\n*n*Results*\n  \n  result:
  settingName:
  ${evidences.key}\n  ${actualValue}\n
Yours
```
Sincerely,
Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "Sample Pager Action",
"pagerdutyServiceKey": "c391356a9d7d4c6b8a0257ff91cc3842",
"pagerdutyEventType": "trigger",
"activeRules": 0,
"disabledRules": 0
},
{
"id": "36bc5690-6dcc-11ea-97c4-57de4ff3eb79",
"name": "Azure Action",
"description": "Azure Action",
"actionType": "qemail",
"createdBy": "John Doe",
"createdById": "user_john",
"updatedBy": "John Doe",
"updatedById": "user_john",
"created": "2020-03-24T12:37:24.729+0000",
"updated": "2020-03-24T12:37:24.729+0000",
"alert": "Qualys CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

Affected Resource
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}

cloudType:${provider.type}
	account:${account.id}

tenantId:${tenantId}

tenantName:${tenantName}

Evaluation Summary
	controlName:${control.name}
	policyName:${policyName}

evaluatedOn:${evaluatedOn}

tfirstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

Results
	result:${control.result}

evidences:
	settingName:${settingName}

evaluatedValue:${evaluatedValue}

Yours
Sincerely,
Qualys Support Team

For any assistance, please contact our customer support team.

"subject": "Azure CV Test",
"smtpHost": "mta01.eng.abc01.example.com",
"smtpPort": 25,
"emailRecipients": [
    "abc@example.com"
],
"emailFromAddress": "noreply@example.com",
"emailReplyTo": "noreply@example.com",
"activeRules": 0,
"disabledRules": 0
},
{
"id": "1f695df0-6da2-11ea-8910-77b847f40d61",
"name": "E-mail Action",
"description": "E-mail Action",
"actionType": "qemail",
"createdBy": "Peter"
"name": "Sample Slack",
"description": "Sample Slack decription",
"actionType": "slack",
"createdBy": "John Doe",
"createdById": "user_john",
"updatedBy": "John Doe",
"updatedById": "user_john",
"created": "2020-03-24T07:36:06.735+0000",
"updated": "2020-03-30T07:54:43.371+0000",
"alert": "Qualys CloudView: Cloud Security Assessment Alerts

*${control.criticality} Severity Control Failure Detected for CID ${cid}*

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}
	cloudType:${provider.type}
	accOUNTID:${account.id}
	tconnectorId:${connectorUuid}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}
	evaluationDates:
		firstEvaluated:${firstEvaluated}
		lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}
	evidences:
		settingName:${evidences.key}
	
tactualValue: ${evidences.value}

Yours Sincerely,
Qualys Support Team

For any assistance, please contact our <mailto:support@qualys.com | customer support team.>",

"slackWebhookUri": "https://hooks.slack.com/services/T95RLRTSL/BRD8PB06QZYxrBEIEx6Mh0R6mMmp1",

"slackChannel": "Sample-slack",
"activeRules": 1,
"disabledRules": 0
}
Get Activities by Id

/rest/v1/activities/{activityId}

[GET]

You can get the list of activities using this API. You can search for activities using filters based on criteria you want.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activityId</td>
<td>(mandatory) (integer) Specify the action ID of an activity in the user's scope.</td>
</tr>
</tbody>
</table>

**Sample - Get activity details using the activity Id**

Let us fetch details of a Slack action using the action Id.

**API request**

```
curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/activities/72652f80-702b-11ea-80dd-7d15cdc80752'
```

**Response**

```
{
    "actionId": "1f695df0-6da2-11ea-8910-77b847f40d61",
    "actionName": "Sample Slack Action",
    "actionType": "slack",
    "aggregate": true,
    "alert": "Qualys CloudView: Cloud Security Assessment Alerts\n    MEDIUM Severity Control Failure Detected for CID 99*\n    Affected Resource:\n    resourceId:arn:aws:lambda:us-east-1:205767712438:function:getEntitlementLambdaEast\n    resourceType:LAMBD\n    service:LAMBDA\n    region:us-east-\n    cloudType:AWS\n    account:205767712438\n    connectorId:f8c3b440-4eaf-11ea-bfef-0dd8ca3bcd2e\n    Evaluation Summary*\n    controlName:Ensure that Multiple Triggers are not configured in $Latest Lambda
```
Function

tcontrolId:99

tpolicyName:AWS Lambda Best Practices

tevaluatedOn:1585314208935

tevaluationDates:

tfirstEvaluated:1581631616942

tlastEvaluated:1585314208935

*Results*
	result: FAIL

evidences:
	settingName:[Total Triggers , Multiple Triggers, Function Arn, Function Name, Role Arn]

Yours Sincerely,
Qualys Support Team

For any assistance, please contact our <mailto:support@qualys.com | customer support team.>

"createdBy": "John Doe",
"createdById": "user_john",
"cloudType": "AWS",
"emailRecipients": [],
"id": "72652f80-702b-11ea-80dd-7d15cdc80752",
"ruleDescription": "Slack 1",
"ruleId": "3dfc5050-7028-11ea-beeb-3fadb6f6b5",
"ruleName": "slack 01 aws",
"slackChannel": "Sample-slack-channel",
"status": "SUCCESS",
"statusDate": 1585314249790,
"isRuleDeleted": false
Reference: Notification Filters

action.message

**ruleName**

Use *quotes or backticks* within values to find rules with certain name.

*Examples*

Find rules with name

**ruleName**: *my first rule*

Find rules that contain parts of the name

**ruleName**: "*my first rule*"

Find rules that match exact value

**ruleName**: `*my first rule*`

**ruleDescription**

Use *quotes or backticks* within values to find rules with certain description.

*Examples*

Find rules with description

**ruleDescription**: *this rule is used for alerting*

Find rules that contain parts of the description

**ruleDescription**: "*this rule is used for alerting*"

Find rules that match exact value

**ruleDescription**: `*this rule is used for alerting*`

**status**

Use a *text value ######* to find rules with certain status (Success, Retrying or Error).

*Example*
Find rules with status

```
status: SUCCESS
```

### statusDate

Use a **date range** or specific date to find when rule status were last modified from one status to another (eg., from Error to Success).

**Examples**

Show rule status modified within certain dates

```
statusDate: [2018-02-01 ... 2018-02-12]
```

Show rule status modified starting 2018-02-01, ending 1 month ago

```
statusDate: [2018-02-01 ... now-1M]
```

Show rule status modified starting 2 weeks ago, ending 1 second ago

```
statusDate: [now-2w ... now-1s]
```

Show rule status modified on certain date

```
statusDate:'2018-02-22'
```

### aggregate

Use the values `true` | `false` to find rules configured to aggregate multiple matches into a single output.

**Example**

Show aggregated rules

```
aggregate: TRUE
```

### createdBy

Use **quotes or backticks** within values to find rules created by a certain user.

**Examples**

Find rules created by user

```
createdBy: Joe Smith
```
Find rules that contain parts of the user name

\texttt{createdBy: "Joe Smith"}

Find rules that contain exact value

\texttt{createdBy: 'Joe Smith'}

\texttt{createdById}

Use \texttt{quotes} or \texttt{backticks} within values to find rules created by a certain user ID.

\textit{Example}

Find rules created by user ID

\texttt{createdById: jsmith}

Find rules that contain parts of the user ID

\texttt{createdById: "jsmith"}

Find rules that match exact value

\texttt{createdById: 'jsmith'}

\texttt{action.name}

Use \texttt{quotes} or \texttt{backticks} within values to find actions with certain name.

\textit{Examples}

Find actions with name

\texttt{action.name: Post to Slack Channel}

Find actions that contain parts of the name

\texttt{action.name: "Post to Slack Channel"}

Find actions that match exact value

\texttt{action.name: 'Post to Slack Channel'}

\texttt{action.type}
Use a **text value ######** to find actions with certain type (Email, slack or pagerduty).

*Example*

Find actions of type

```
action.type: SLACK
```

*action.message*

Use **quotes or backticks** within values to find rules with certain text in the message (email, slack or pagerduty messages).

*Examples*

Find rules with message

```
action.message: to operations team
```

Find rules that contain parts of the message

```
action.message: "to operations team"
```

Find rules that match exact value

```
action.message: `to operations team`
```

*action.emailRecipient*

Use **quotes or backticks** within values to find actions with certain email recipients.

*Examples*

Find actions with email recipient

```
action.emailRecipient: secops-alert@mycompany.com
```

Find actions that contain parts of the email recipient

```
action.emailRecipient: "secops-alert@mycompany.com"
```

Find actions that match exact value

```
action.emailRecipient: `secops-alert@mycompany.com`
```
**action.subject**

Use **quotes or backticks** within values to find actions with certain text in the subject (email or pagerduty subject).

**Examples**

Find actions with subject

```
action.subject: warning
```

Find actions that contain parts of the subject

```
action.subject: "warning"
```

Find actions that match exact value

```
action.subject: `warning`
```

**action.slackChannel**

Use **quotes or backticks** within values to find actions with certain slack channel name.

**Examples**

Find actions with slack channel

```
action.slackChannel: Sec Ops
```

Find actions that contain parts of the slack channel name

```
action.slackChannel: "Sec Ops"
```

Find actions that match exact value

```
action.slackChannel: `Sec Ops`
```
Response Rules

We provide APIs to create rule, update a rule, delete a rule, enable or disable rules. Before you proceed with creation of rules, ensure that you have pre-defined actions for the rule.

Get Rules

Get Rules by Id

Create Rules

Update Rules

Delete Rules

Disable Rules

Enable Rules
CloudView APIs

Get Rules
/rest/v1/rules

[GET]

You can get the list of rules using this API. You can search for rules for a cloud provider using filters we support

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloudType</td>
<td>(mandatory) Select the cloud provider from AWS, Azure, or GCP.</td>
</tr>
<tr>
<td>filter</td>
<td>Form the search query using the filters we provide to refine the search for rules.</td>
</tr>
</tbody>
</table>

Filters supported:

- ruleName
- rule.description
- trigger
- ruleQuery
- createdBy
- createdById
- updatedBy
- updatedById
- ruleState
- createdDate
- updatedDate
- lastRun
- aggregate
- aggregationGroup
- action.message
- action.subject
- action.slackChannel
- action.emailRecipient
- action.type
- action.name

For detailed information on filters, see the Reference:
**CloudView APIs**

**Action Filters.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
<tr>
<td>sortField</td>
<td>Specify the field that decides the sort order for the rules.</td>
</tr>
<tr>
<td>sortOrder</td>
<td>{asc</td>
</tr>
</tbody>
</table>

**Sample - Get the list of rules**

Let us get the rules for simple_alerts rule type for AWS cloud provider.

**API request**

curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXN1cm5hbWVu6cGFzc3dvcmQK==' 'https://<QualysURL>/cloudview-api/rest/v1/rules?cloudType=AWS&pageNo=1&pageSize=50&ruleType=simple_alert&sortOrder=asc'

**Response**

```json
[
  {
    "id": "3dfc5050-7028-11ea-beeb-3fad76b6f6b5",
    "clouIdType": "AWS",
    "ruleType": "simple_alert",
    "name": "slack 01 aws",
    "description": "Slack 1",
    "qql": "cid:99 and account.id:XXXXXXXXXXXXX and control.result:FAIL",
    "aggregate": false,
    "actions": [
      {
        "id": "1f695df0-6da2-11ea-8910-77b847f40d61",
        "actionType": "slack",
        "name": "slack cv public api",
        "subject": null,
```
CloudView APIs

"alert": "Qualys CloudView: Cloud Security Assessment Alerts

*{$control.criticality} Severity Control Failure Detected for CID ${cid}*

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}

tcloudType:${provider.type}

taccountId:${account.id}

tconnectorId:${connectorUuid}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}
	evaluatedOn:${evaluatedOn}

evaluationDates:
	firstEvaluated: ${firstEvaluated}
	lastEvaluated: ${lastEvaluated}

*Results*
	result: ${control.result}
	evidences:
		settingName:${evidences.key}
	actualValue: ${evidences.value}

Yours Sincerely,
Qualys Support Team

For any assistance, please contact our customer support team.",

"emailRecipients": null,
"slackChannel": "Sample-slack",
"subjectParameters": [],
"bodyParameters": []
}

,"created": "2020-03-27T12:41:12.917+0000",
"createdBy": "John Doe",
"createdById": "user_john",
"updated": "2020-03-27T12:41:12.917+0000",
"updatedBy": "John Doe",
"updatedById": "user_john",
"lastRun": "2020-04-29T05:39:32.974+0000",
"ruleState": "DISABLED",
"durationHour": 0,
"fromHourInUTC": 0,
"fromMinuteInUTC": 0
}

{ "id": "368fea00-702a-11ea-beeb-3fad76b6f6b5",
"cloudType": "AWS",
"ruleType": "time_window_schedule_alert",
"days": [1, 2, 3, 4, 5, 6, 7]
CloudView APIs
],
"name": "time window",
"description": "Time",
"qql": "cid:98 and accountGroup:Sample and control.result:FAIL",
"aggregate": true,
"aggregationKey": "account.id",
"actions": [
{
"id": "2a8bda80-7029-11ea-beeb-3fad76b6f6b5",
"actionType": "qemail",
"name": "Time email",
"subject": "Time window",
"alert": "Qualys CloudView: Cloud Security Assessment
Alerts\n\n${control.criticality} Severity Control Failure Detected for
CID ${cid}\n\n*Affected
Resource*\n\tresourceId:${resource.id}\n\tresourceType:${resource.type
}\n\tservice:${service.type}\n\tregion:${region}\n\tcloudType:${provid
er.type}\n\taccountId:${account.id}\n\tconnectorId:${connectorUuid}\n\
tgroupName:${accountGroup}\n\n*Evaluation
Summary*\n\tcontrolName:${control.name}\n\tcontrolId:${cid}\n\tpolicyN
ame:${policyName}\n\tevaluatedOn:${evaluatedOn}\n\tevaluationDates:\n\
t\tfirstEvaluated:${firstEvaluated}\n\t\tlastEvaluated:${lastEvaluated
}\n\n*Results*\n\tresult:${control.result}\n\tevidences:\n\t\tsettingN
ame:${evidences.key}\n\t\tactualValue:${evidences.value}\n\nYours
Sincerely,\nQualys Support Team\n\n\nFor any assistance, please
contact our customer support team.",
"emailRecipients": [
"abc@example.com"
],
"slackChannel": null,
"subjectParameters": [],
"bodyParameters": []
}
],
"created": "2020-03-27T12:55:19.456+0000",
"createdBy": "John Doe",
"createdById": "user_john",
"updatedBy": "John Doe",
"updatedById": "user_john",
"lastRun": "2020-03-27T14:00:00.163+0000",
"ruleState": "DISABLED",
"durationHour": 3600000,
"fromHourInUTC": 13,
"fromMinuteInUTC": 0
},
209


{ "id": "12ec9a00-7028-11ea-beeb-3fad76b6f6b5", "cloudType": "AWS", "ruleType": "simple_alert", "name": "test01 aws", "description": "Test1", "qql": "cid:100 and account.id:XXXXXXXXXXXXand control.result:FAIL and firstEvaluated:[now-1M .. now]", "aggregate": false, "actions": [ { "id": "f913b4a0-6d9e-11ea-97c4-57de4ff3eb79", "actionType": "qemail", "name": "Public ApI", "subject": "Public API testing", "alert": "Qualys CloudView: Cloud Security Assessment Alerts

${control.criticality} Severity Control Failure Detected for CID ${cid}

*Affected Resource*
	resourceId:${resource.id}
	resourceType:${resource.type}
	service:${service.type}
	region:${region}

tcloudType:${provider.type}
	accountId:${account.id}

tconnectorId:${connectorUuid}

tgroupName:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	controlId:${cid}
	policyName:${policyName}

evaluatedOn:${evaluatedOn}

tevaluationDates:
	firstEvaluated:${firstEvaluated}
	lastEvaluated:${lastEvaluated}

*Results*
	result:${control.result}

evidences:
	settingName:${evidences.key}
	actualValue:${evidences.value}

Yours
Sincerely,
Qualys Support Team

For any assistance, please contact our customer support team.", "emailRecipients": [ "abc@example.com" ], "slackChannel": null, "subjectParameters": [], "bodyParameters": [] }, "created": "2020-03-27T12:40:00.672+0000", "createdBy": "John Doe", "createdById": "user_john", "updated": "2020-03-27T13:04:03.135+0000", "updatedBy": "John Doe", "updatedById": "user_john", "lastRun": "2020-03-27T13:04:03.135+0000", "ruleState": "DISABLED", "durationHour": 0,}
"fromHourInUTC": 0,
"fromMinuteInUTC": 0
},
{
"id": "dcf05f80-8ad1-11ea-9f4c-35b43d39dafc",
"cloudType": "AWS",
"ruleType": "simple_alert",
"name": "slack New template rule 01",
"description": "slack New Template",
"qql": "cid:99 and account.id:XXXXXXXXXXXXXand
control.result:FAIL",
"aggregate": false,
"actions": [

{
"id": "51cba540-8ad1-11ea-9f4c-35b43d39dafc",
"actionType": "slack",
"name": "slack new template",
"subject": null,
"alert": "Qualys CloudView: Cloud Security Assessment
Alerts
An assessment failure has been identified for resource
"${resource.id}" and control "${cid}" in your Qualys
subscription.

*Impacted
Resource*
resourceId:${resource.id}
resourceType:${resource.type}
service:${service.type}
region:${region}
cloudType:${provider.type}
accountId:${account.id}
tenantId:${connectorUuid}
groupName:${accountGroup}

*Evaluation
Summary*
controlName:${control.name}
cid:${cid}
policyName:${policyName}
evaluatedOn:${evaluatedOnDateFormat}
evaluationDates:
firstEvaluated:${firstEvaluatedDateFormat}
lastEvaluated:${lastEvaluatedDateFormat}
Evidence*:
result:${control.result}
evidences:
settingName:${evidences.key}
actualValue:${evidences.value}

Use this information here to investigate the
failure and take appropriate actions to fix it.

"emailRecipients": null,
"slackChannel": "Sample-slack",
"subjectParameters": [],
"bodyParameters": []
}

"created": "2020-04-30T11:00:54.776+0000",
"createdBy": "John Doe",
"createdById": "user_john",
"updated": "2020-04-30T11:00:54.776+0000",
"updatedBy": "John Doe",
"updatedById": "user_john",
"lastRun": "2020-04-30T11:10:36.749+0000"
"ruleState": "ENABLED",
"durationHour": 0,
"fromHourInUTC": 0,
"fromMinuteInUTC": 0
}
Get Rules by Id

/rest/v1/rules/{ruleId}

[GET]

Specify the rule ID and fetch rule details.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ruleId</td>
<td>(mandatory) (integer) Specify the ID of rule in the user’s scope.</td>
</tr>
</tbody>
</table>

**Sample - Get rule details using the rule Id**

Let us get the rules for simple_alerts rule type for AWS cloud provider.

**API request**

```bash
curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQK==' 'https://<QualysURL>/cloudview-api/rest/v1/rules/368fea00-702a-11ea-beeb-3fad76b6f6b5'
```

**Response**

```json
{
    "id": "368fea00-702a-11ea-beeb-3fad76b6f6b5",
    "cloudType": "AWS",
    "ruleType": "time_window_schedule_alert",
    "days": [1, 2, 3, 4, 5, 6, 7]
}
```
"name": "time window",
"description": "Time",
"qql": "cid:98 and accountGroup:sampleaccount and control.result:FAIL",
"aggregate": true,
"aggregationKey": "account.id",
"actions": [
{
"id": "2a8bda80-7029-11ea-beeb-3fad76b6f6b5",
"actionType": "qemail",
"name": "Time email",
"subject": "Time window",
"alert": "Qualys CloudView: Cloud Security Assessment Alerts\n\n${control.criticality} Severity Control Failure Detected for CID ${cid}\n\n*Affected Resource*\n	resourceId:${resource.id}\n	resourceType:${resource.type}\n	service:${service.type}\n	region:${region}\n	cloudType:${provider.type}\n	accountId:${account.id}\n	connectorId:${connectorUuid}\n	accountGroupName:${accountGroup}\n\n*Evaluation Summary*\n	controlName:${control.name}\n	controlId:${cid}\n	policyName:${policyName}\n	evaluatedOn:${evaluatedOn}\n	evaluationDates:\n		firstEvaluated:${firstEvaluated}\n		lastEvaluated:${lastEvaluated}\n\n*Results*\n	result:${control.result}\n	evidences:\n		settingName:${evidences.key}\n		actualValue:${evidences.value}\n\nYours Sincerely,
Qualys Support Team\n\nFor any assistance, please contact our customer support team.",
"emailRecipients": [
"abc@example.com"
],
"slackChannel": null,
"subjectParameters": [],
"bodyParameters": []
},
"created": "2020-03-27T12:55:19.456+0000",
"createdBy": "John Doe",
"createdById": "user_john",
"updatedBy": "John Doe",
"updatedById": "user_john",
"ruleState": "ENABLED",
"durationHour": 3600000,
"fromHourInUTC": 13,
"fromMinuteInUTC": 0
Create Rules

/rest/v1/rules

[POST]

You can create rule and specify the criteria for the alert to be generated using the actions you define. Specify the necessary details in the request body that are required to create rule such as actionId, actionType, emailRecipients, emailSubject, and so on.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloudType</td>
<td>(mandatory) Select the cloud provider from AWS, Azure, or GCP.</td>
</tr>
</tbody>
</table>
| ruleType     | Select the rule type: simple_alert or time_window_schedule_alert. Depending on the rule type you select, the elements in the ruleBody are different. Select the rule type: simple_alert or time_window_schedule_alert. Depending on the rule type you select, the elements in the ruleBody are different. Simple_alert: For simple_alert rule type, below parameters are optional.  
  
aggregate  
aggregationKey  
durationHour  
fromHourInUTC  
fromMinuteInUTC  

  
time_window_schedule_alert: For time_window_schedule_alert rule type, you need to provide all the parameters. |
(body) Specify the different elements needed in the request body for a rule. Refer to the following example for exact syntax.

```json
{
    "actionRequests": [
        {
            "actionId": "string",
            "actionType": "qemail",
            "emailRecipients": [
                "string"
            ],
            "emailSubject": "string",
            "pagerSubjectLine": "string",
            "slackChannel": "string"
        }
    ],
    "aggregate": true,
    "aggregationKey": "string",
    "description": "string",
    "durationHour": 0,
    "fromHourInUTC": 0,
    "fromMinuteInUTC": 0,
    "name": "string",
    "qql": "string"
}
```

Where,

- **actionId**: ID of the action you have defined.
- **actionType**: type of the action to be implemented: qemail, pagerduty, or slack.
- **emailRecipients**: valid email ID of the recipients to whom the alert should be sent. You can provide multiple email IDs separated by comma.

Depending on the application mode you choose to send alerts, you may define either one or more elements:

- **emailSubject**: subject of the email action
- **pagerSubjectLine**: subject for alert using PagerDuty
application

slackChannel: name of the channel to access Slack application

Sample - Create a rule using Slack application

API request

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/rules?cloudType=AWS&ruleType=time_window_schedule_alert'

Request POST Data

{
   "actionRequests": [
      {
         "actionId": "b2af9830-5dfe-11ea-b157-8ba65cd99c15",
         "actionType": "slack"
      }
   ],
   "aggregate": true,
   "aggregationKey": "region",
   "description": "Slack Public API Rule",
   "durationHour": 0,
   "fromHourInUTC": 0,
   "fromMinuteInUTC": 0,
   "name": "Slack Api",
   "qql": "cid:99 and account.id:XXXXXXXXXXXX and control.result:FAIL and firstEvaluated:[now-4M .. now]"
}

Response

{
   "success": "5ac209e0-9966-11e8-ab43-6187ace8f6e8"
}
Update Rules

/rest/v1/rules/{ruleId}

[PUT]

You can update rules. Specify the necessary details in the request body that are required to update an rules such as action ID, action name, action description, the recipient details, whom the email should be sent to and so on.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ruleId</td>
<td>(mandatory) (integer) Specify the ID of rule in the user’s scope.</td>
</tr>
<tr>
<td>ruleBody</td>
<td>(body) Specify the different elements needed in the request body for a rule. Refer to the following example for exact syntax.</td>
</tr>
</tbody>
</table>

```json
{
    "actionRequests": [
    {
        "actionId": "string",
        "actionType": "qemail",
        "emailRecipients": [
            "string"
        ],
        "emailSubject": "string",
        "pagerSubjectLine": "string",
        "slackChannel": "string"
    }
    ],
    "aggregate": true,
    "aggregationKey": "string",
    "description": "string",
    "durationHour": 0,
    "fromHourInUTC": 0,
    "fromMinuteInUTC": 0,
    "name": "string",
    "qql": "string"

```
Where,

actionId: ID of the action you have defined.

actionType: type of the action to be implemented: qemail, pagerduty, or slack.

emailRecipients: valid email ID of the recipients to whom the alert should be sent. You can provide multiple email IDs separated by comma.

Depending on the application mode you choose to send alerts, you may define either one or more elements:

emailSubject: subject of the email action

pagerSubjectLine: subject for alert using PagerDuty application

slackChannel: channel name to access Slack application

Sample - Update rules

API request

curl -k -X PUT -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/rules/1a841990-5dff-11ea-a923-6b29e6c4cbec?ruleType=simple_alert'

Request PUT Data

```
{
  "actionRequests": [
    {
      "actionId": "b2af9830-5dfe-11ea-b157-8ba65cd99c15",
      "actionType": "slack"
    }
  ],
  "aggregate": true,
  "aggregationKey": "region",
  "description": "Slack Public API Rule",
  "durationHour": 0,
```
```
"fromHourInUTC": 0,
"fromMinuteInUTC": 0,
"name": "Slack Api",
"qql": "cid:99 and account.id:205767712438 and control.result:FAIL
and firstEvaluated:[now-4M .. now]"
}

Response
{
    "success": "bd786210-9965-11e8-ab43-6187ace8f6e8"
}
```
Delete Rules

/rest/v1/rules/delete

[POST]

Specify the ID of an existing rule you want to delete and the rule gets deleted. Ensure that the rules you want to delete are disabled. If a rule is enabled, you cannot delete the rule.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ruleIds</td>
<td>(Array [string]) Specify the ID of an rule to be deleted and the rule gets deleted. You can provide multiple Ids separated by comma.</td>
</tr>
</tbody>
</table>

Example:

```
{
  "ids": [
    "string"
  ]
}
```

**Sample - Delete rules**

**API request**

```
curl -k -X POST-u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/rules/delete'
```

**Request POST Data**

```
{
  "ids": [
    "1a841990-5dff-11ea-a923-6b29e6c4cbec",
    "efbf4080-52dd-11ea-a008-cbe911ab6a51"
  ]
}
```
Response

No Content
Response Code: 200

Response returns status code 200 with rule detail for rules that cannot be deleted. If rules are in enabled state, you need to disable them before deleting.

```json
{
  "efbf4080-52dd-11ea-a008-cbe911ab6a51": "Cannot delete enable rule."
}
```
Disable Rules

/rest/v1/rules/disable

[POST]

Specify the ID of an existing rule you want to disable and the rule gets disabled.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ruleIds</td>
<td>(Array [string]) Specify the ID of an rule to be deleted and the rule gets disabled. You can provide multiple Ids separated by comma. Example:</td>
</tr>
<tr>
<td></td>
<td>{ &quot;ids&quot;: [ &quot;string&quot; ] }</td>
</tr>
</tbody>
</table>

**Sample - Disable rules**

**API request**

curl -k -X POST-u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/rules/disable'

**Request POST Data**

```json
{
    "ids": [
        "1a841990-5dff-11ea-a923-6b29e6c4cbec",
        "efbf4080-52dd-11ea-a008-cbe911ab6a51"
    ]
}
```
<table>
<thead>
<tr>
<th>Response</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No Content</td>
<td></td>
</tr>
<tr>
<td>Response Code: 200</td>
<td></td>
</tr>
</tbody>
</table>
Enable Rules

/rest/v1/rules/enable

[POST]

Specify the ID of an existing rule you want to enable and the rule gets enabled.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ruleIds</td>
<td>(Array [string]) Specify the ID of an rule to be enabled and the rule gets enabled. You can provide multiple Ids separated by comma.</td>
</tr>
</tbody>
</table>

Example:

```json
{
   "ids": [
      "string"
   ]
}
```

**Sample - Enable rules**

**API request**

```
curl -k -X POST-u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/rules/enable'
```

**Request POST Data**

```json
{
   "ids": [
      "1a841990-5dff-11ea-a923-6b29e6c4cbec",
      "efbf4080-52dd-11ea-a008-cbe911ab6a51"
   ]
}
```

**Response**
CloudView APIs

No Content
Response Code: 200
Reference: Rule Filters

You can form the search query using the filters we provide to refine the search for actions.

**ruleName**

Use **quotes or backticks** within values to find rules with certain name.

*Examples*

Find rules with name

```python
ruleName: my first rule
```

Find rules that contain parts of the name

```python
ruleName: "my first rule"
```

Find rules that match exact value

```python
ruleName: 'my first rule'
```

**ruleDescription**

Use **quotes or backticks** within values to find rules with certain description.

*Examples*

Find rules with description

```python
ruleDescription: this rule is used for alerting
```

Find rules that contain parts of the description

```python
ruleDescription: "this rule is used for alerting"
```

Find rules that match exact value

```python
ruleDescription: 'this rule is used for alerting'
```

**trigger**

Use a **text value ** to find rules with a certain trigger (Single Match or Time Window Scheduled Match).
Example

Find rules with trigger

```
trigger: SINGLE MATCH
```

Use **quotes or backticks** within values to find rules with a certain query (use Qualys Query Language).

**Examples**

Find rules with query

```
ruleQuery: asset.score
```

Find rules that contain parts of the query

```
ruleQuery: "asset.score"
```

Find rules that match exact value

```
ruleQuery: `asset.score`
```

**createdBy**

Use **quotes or backticks** within values to find rules created by a certain user.

**Examples**

Find rules created by user

```
createdBy: Joe Smith
```

Find rules that contain parts of the user name

```
createdBy: "Joe Smith"
```

Find rules that match exact value

```
createdBy: `Joe Smith`
```

**createdById**

Use **quotes or backticks** within values to find rules created by a certain user ID.
Example

Find rules created by user ID

createdById: jsmith

Find rules that contain parts of the user ID

createdById: "jsmith"

Find rules that match exact value

createdById: 'jsmith'

ruleState

Use a text value # to find rules by a certain running state (Enabled or Disabled).

Example

Find rules with state

ruleState: ENABLED

createdDate

Use a date range or specific date to find when rules were created.

Examples

Show rules created within certain dates

createdDate: [2018-02-01 ... 2018-02-12]

Show rules created starting 2018-02-01, ending 1 month ago

createdDate: [2018-02-01 ... now-1M]

Show rules created starting 2 weeks ago, ending 1 second ago

createdDate: [now-2w ... now-1s]

Show rules created on certain date

createdDate: '2018-02-22'
updatedDate

Use a **date range** or specific date to find when rules were last modified.

*Examples*

Show rules updated within certain dates

`updatedDate: [2018-02-01 ... 2018-02-12]`

Show rules updated starting 2018-02-01, ending 1 month ago

`updatedDate: [2018-02-01 ... now-1M]`

Show rules updated starting 2 weeks ago, ending 1 second ago

`updatedDate: [now-2w ... now-1s]`

Show rules updated on certain date

`updatedDate: '2018-02-22'`

lastRun

Use a **date range** or specific date to find when rules were last executed.

*Examples*

Show rules last run within certain dates

`lastRun: [2018-02-01 ... 2018-02-12]`

Show rules last run starting 2018-02-01, ending 1 month ago

`lastRun: [2018-02-01 ... now-1M]`

Show rules last run starting 2 weeks ago, ending 1 second ago

`lastRun: [now-2w ... now-1s]`

Show rules last run on certain date

`lastRun: '2018-02-22'`
Use the values `true` | `false` to find rules configured to aggregate multiple matches into a single output.

**Example**

Show aggregated rules

```yaml
aggregate: True
```

**aggregationGroup**

Use quotes or backticks within values to find rules aggregated into a certain group.

**Examples**

Find rules with aggregation group

```yaml
aggregationGroup: hostname
```

Find rules that contain parts of the aggregation group name

```yaml
aggregationGroup: "hostname"
```

Find rules that match exact value

```yaml
aggregationGroup: 'hostname`
```

**action.message**

Use quotes or backticks within values to find rules with certain text in the message (email, slack or pagerduty messages).

**Examples**

Find rules with message

```yaml
action.message: to operations team
```

Find rules that contain parts of the message

```yaml
action.message: "to operations team"
```

Find rules that match exact value

```yaml
action.message: `to operations team`
```
action.subject

Use *quotes or backticks* within values to find rules with certain text in the subject (email or pagerduty subject).

**Examples**

Find rules with subject

```python
action.subject: warning
```

Find rules that contain parts of the subject

```python
action.subject: "warning"
```

Find rules that match exact value

```python
action.subject: `warning`
```

action.slackChannel

Use *quotes or backticks* within values to find rules with certain slack channel name.

**Examples**

Find rules with slack channel

```python
action.slackChannel: Sec Ops
```

Find rules that contain parts of the slack channel name

```python
action.slackChannel: "Sec Ops"
```

Find rules that match exact value

```python
action.slackChannel: `Sec Ops`
```

action.emailRecipient

Use *quotes or backticks* within values to find rules with certain email recipients.

**Examples**

Find rules with email recipient
**action.emailRecipient**: secops-alert@mycompany.com

Find rule that contain parts of the email recipient

**action.emailRecipient**: "secops-alert@mycompany.com"

Find rules that match exact value

**action.emailRecipient**: `secops-alert@mycompany.com`

**action.type**

Use a text value ####### to find rules with certain action type (Email, slack or pagerduty).

*Example*

Find rules of action type

**action.type**: EMAIL

**action.name**

Use quotes or backticks within values to find rules with certain action name.

*Examples*

Find rules with action

**action.name**: Post to Slack Channel

Find rules that contain parts of the action name

**action.name**: "Post to Slack Channel"

Find rules that match exact value

**action.name**: `Post to Slack Channel`
User Access Management APIs

Get the User Scope

/rest/v1/users/{userName}/scope

[GET]

You can fetch the group details by specifying the unique Id assigned to a group.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>(string) Provide the username for which the scope needs to be determined.</td>
</tr>
</tbody>
</table>

Sample - Get the list of groups

**API request**

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/users/user_john/scope'

**Response**

```
"groups": [
  {
    "uuid": "52660405-27d3-3f69-b764-b0061ab4c494",
    "title": "Example_group",
    "connectorCount": 2
  }
],
"AWS": {
  "directAccountScope": [
    {
      "connectorUuid": "af50f5c0-c8c2-11e9-945e-77a38645daea",
      "accountIdentifier": "XXXXXXXXXXXX",
      "cloudType": "AWS",
```
"connectorName": "AWS_Connector_1"
],
"regions": [
"us-east-1",
"us-east-2"
]
},
"AZURE": {
"directAccountScope": [
{
"connectorUuid": "2e0c1660-d061-11e9-ad71-df4fba75b3c5",
"accountIdentifier": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX",
"cloudType": "AZURE",
"connectorName": "Azure_Connector_2"
}
]
},
"GCP": {
"directAccountScope": [
{
"connectorUuid": "66e7d1f0-c8c2-11e9-9fcb-85661d3ad949",
"accountIdentifier": "gcp-demo",
"cloudType": "GCP",
"connectorName": "GCP_Connector_3"
}
]
}
Update Groups Scope for User

/rest/v1/users/{userName}/groupScope

[POST]

You can now update the groups associated with a specific user. You could add new groups and remove groups that are associated with the user using update operation.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>Provide the username for which the group scope needs to be updated.</td>
</tr>
<tr>
<td>userGroupModifyRequest</td>
<td>(body) Use this to specify the group IDs that you want to add and remove.</td>
</tr>
</tbody>
</table>

Example:

```json
{
    "add": {
        "groupsIds": ["string"
    },
    "remove": {
        "groupsIds": ["string"
    }
}
```

where,

groupIds: unique ID assigned to the group.

Example: ea4b240f-c27c-30a6-ba28-8fc9a38fa8d1

Sample - Update Connectors in a group

**API request**

```bash
curl -k -X POST -u <username>:<password>
```
Request POST Data

userGroupModifyRequest:
{
  "add": {
    "groupsIds": ["52660405-27d3-3f69-b764-b0061ab4c494"]
  },
  "remove": {
    "groupsIds": ["9d665fd0-f15d-379f-8b11-b39cd4ebfd9e"]
  }
}

Response

{
  "groups": [
    {
      "uuid": "52660405-27d3-3f69-b764-b0061ab4c494",
      "title": "new-group",
      "connectorCount": 2
    }
  ],
  "AWS": {
    "directAccountScope": [],
    "regions": []
  },
  "AZURE": {
    "directAccountScope": []
  },
  "GCP": {
    "directAccountScope": []
  }
}
Update Connector Scope for user

/rest/v1/users/{userName}/scope

[POST]

You can now update the connectors associated with a specific connector. You could add new groups and remove groups that are associated with the connector using update operation.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>Provide the username for which the group scope needs to be updated.</td>
</tr>
<tr>
<td>userGroupModifyRequest</td>
<td>Use this to specify the group IDs that you want to add and remove.</td>
</tr>
</tbody>
</table>

Example:

```json
{
  "add": {
    "accountIdentifiers": ["string"],
    "regions": ["string"]
  },
  "remove": {
    "accountIdentifiers": ["string"],
    "regions": ["string"]
  }
}
```

where,

accountIdentifiers: The unique identifier associated with a connector. For every cloud provider, the identifier is different.

- AWS: account ID (Example: 11111111111)
CloudView APIs

- Azure: subscription ID (Example: 11111111-1111-1111-1111-111111111111)
- GCP: project ID (Example: sample_gcp)

regions: (applicable only for AWS connectors)
Specify the region of the connector

### cloudType
Select the cloud provider of the connector being updated: AWS, Azure or GCP.

Sample - Update the Groups associated with the Connector

#### API request

curl -k -X POST -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/users/user_john/scope?cloudType=AWS'

#### Request POST Data

```json
{
  "add": {
    "accountIdentifiers": [
      "XXXXXXXXXXXX"
    ],
    "regions" : ["us-east-1","us-east-2"]
  },
  "remove": {
    "accountIdentifiers": [
      "XXXXXXXXXXXX"
    ],
    "regions" : ["eu-west-1"]
  }
}
```

CloudType: AWS

#### Response

```json
{
  "groups": [
    {
      "uuid": "52660405-27d3-3f69-b764-b0061ab4c494",
      "title": "new_sample_group"
    }
  ]
}
```
"connectorCount": 2
},
"AWS": {
"directAccountScope": [
{
"connectorUuid": "af50f5c0-c8c2-11e9-945e-77a38645daea",
"accountIdentifier": "XXXXXXXXXXXX",
"cloudType": "AWS",
"connectorName": "AWS_Connector"
}
],
"regions": [
"us-east-1",
"us-east-2"
]
},
"AZURE": {
"directAccountScope": []
},
"GCP": {
"directAccountScope": [
{
"connectorUuid": "66e7d1f0-c8c2-11e9-9fcb-85661d3ad949",
"accountIdentifier": "gcp-demo",
"cloudType": "GCP",
"connectorName": "GCP_Connector"
}
]
}
Secure IaC

Secure Infrastructure as Code

In the current continuous integration and continuous deployment (CICD) environment, the scans are conducted on cloud resources after deployment. As a result, you secure the cloud resources post deployment. We are introducing Infrastructure as Code (IaC) Security feature for AWS Terraform. With arrival of IaC scan, you can now secure your code (IaC) before it gets deployed in the cloud environment.

The new Qualys IaC Security feature will help shifting security and compliance posture of cloud security to left, allowing evaluation of cloud resource misconfigurations even before actual deployment. Using this feature, cloud infrastructure teams can prevent misconfigurations before it really happens.

The first step towards IaC security is triggering an IaC scan. In the current scenario, the scans are executed after the cloud resources are deployed in the cloud environment. As a result, fixing of misconfigurations happens post deployment. However, using this feature, you can trigger the scan on IaC (configuration file) before the cloud resources are deployed in the environment.

Once you trigger the scan, we will evaluate the configuration file (IaC) against pre-defined controls.

IaC scanning works by uploading the template file or zip containing multiple files to CloudView, either via our CLI or API. The template is processed, and the response returns a scan ID. The returned scan id then can be used to fetch the scan report which provides the evaluation results giving you a clear picture of the misconfigurations (if any) that need to be fixed to secure your code before the actual deployment.

You can scan the templates either through CLI commands or using APIs:

Scanning Template Files Using CLI

Scanning Template Files Using API

Want to know more?

Pre-requisites
Template Support

Understanding IaC Scan Output
Pre-requisites & Template Support

Template Support

This Qualys IaC Security version supports following template files:

- **AWS, Azure, and GCP Terraform Templates**: The .tf template files - IaC Security scan supports over 100 terraform resource types.

- **AWS, Azure, and GCP Terraform Plan**: The .json plan files - To scan the plan files, you need to make those files available in JSON format. Refer https://www.terraform.io/docs/internals/json-format.html

- **AWS Cloudformation Template**: We support the file types: .json, .yaml, .yml, .template

- **Compressed Template File Formats**: We are supporting following compressed template file format:
  - .zip
  - .7z
  - .tar
  - .tar.gz
  - .gz

Pre-requisite

Users with a non-expired paid/trial version of Cloud Security Assessment (CSA) subscription that has API access enabled. The following users with required permissions can access IaC:

- A user with Manager access

- A sub-user with the CLOUDVIEW API Access
Scanning Template Files Using CLI

Qualys provides a IaC scanning CLI which can be installed on any machines having python3. Qualys IaC Security CLI is based on Python PIP Platform. For complete details, refer to Secure IaC section in CloudView User Guide.
Scanning Template Files Using APIs

Scanning Template Files Using APIs

Qualys provides the following APIs to launch the IaC scan and fetch the scan results and scan lists.

1) Trigger IaC Scan (POST)

2) Get Scan Results (GET)

3) Get List of Scans (GET)
Trigger IaC Scan

/rest/v1/iac/scan

[POST]

You can trigger an IaC scan. Provide a name and upload the IaC configuration file to be scanned. Once the scan is triggered, it goes into Submitted state. Once the scan is completed (Finished state), the response provides a unique Scan UUID that you can use to view the scan results.

Note: We support only 10 concurrent scans to be executed in parallel.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(Required) Provide a name for the IaC scan you would want to trigger. Note: Double quotes are not allowed.</td>
</tr>
<tr>
<td>file</td>
<td>(Required) Upload an IaC configuration file with maximum size of 10MB. For valid file formats, refer to Template Support.</td>
</tr>
<tr>
<td>showOnlyFailedControls</td>
<td>(boolean) Set this flag to true to include only the failed controls in the IaC scan result.</td>
</tr>
<tr>
<td>tags= [{‘key’: ‘value’}]</td>
<td>Name of the tags. The cloud assets are tagged with specified tag are included in the scan.</td>
</tr>
<tr>
<td>policyName</td>
<td>Specify the name of the policy in the request. Use the policy name to restrict the evaluation of controls during the scan. Only the controls associated with the specified policy are evaluated during the scan. If PolicyName parameter is empty, the IaC Scan API scans the template for all the controls that are applicable to resources in the template and return the scan results accordingly.</td>
</tr>
</tbody>
</table>
Sample - Trigger an IaC Scan

**API request**

```bash
curl -X POST
'https://<QualysBaseURL>/cloudview-api/rest/v1/iac/scan'
-H 'authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXX='
-H 'Content-Type: multipart/form-data'
-F 'file=@security-group.tf'
-F 'name=DemoTemplate'
-F 'policyName=AZURE Infrastructure as Code Security Best Practices Policy'
-F 'showOnlyFailedControls=false'
-F 'tags=[{"Key":"Value"}]'
```

**Response**

```json
{
   "scanUuid": "337a21ef-3c53-43bf-aed6-46f04e1c542d"
}
```
Get IaC Scan Results

/rest/v1/iac/scanResult?scanUuid=[id]

[GET]

Use scanUuid returned by Trigger scan API to fetch scan results. The scan results can be fetched only when the IaC scan is completed. If you try to fetch the scan results before it is completed, you will see respective message in response. For example, Scan is in a processing state.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scanUuid</td>
<td>(Required) Unique identifier assigned to the IaC scan. The scan Uuid is returned in the response of Trigger IaC scan.</td>
</tr>
<tr>
<td>responseFormat</td>
<td>Select the format in which the response should be displayed. You can choose from JSON, SARIF format types. By default, response is displayed in JSON format. Note: If you specify SARIF format and the scan is not completed, then the response is in JSON format. SARIF format is returned only for completed scans.</td>
</tr>
</tbody>
</table>

Sample - Trigger an Iac Scan (JSON)

**API request**

```
curl -X GET "https://<QualysBaseURL>/cloudview-api/rest/v1/iac/scanResult?scanUuid=337a21ef-3c53-43bf-aed6-46f04e1c542d" -H 'authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
```

**Response (JSON)**

```
{
    "scanUuid": "337a21ef-3c53-43bf-aed6-46f04e1c542d",
    "scanDate": "2021-06-22T11:13:37.275+00:00",
}
```
Sample_Scan

status: FINISHED

tags: []

result:

- checkType: terraform
  results:
    passedChecks:
      - checkId: CKV_AWS_60
        checkName: Ensure IAM role allows only specific services or principals to assume it
        criticality: HIGH
        cvControl: null
        checkResult:
          result: PASSED
          evaluatedKeys: []

    codeBlock:
      - resource "aws_iam_role" "dynamodb-dax-cluster-iam-role-fail"
        name = "dax-cluster-iam-role-fail"
        resource: aws_dynamodb_table.dynamodb-table-fail

    summary:
      passed: 5
      failed: 3
      failedStats:


We have now added error codes to the response when you fetch IaC scan results for troubleshooting purposes. You can provide the error codes to Qualys support that helps us troubleshoot the issue.

Error Sample - Trigger an IaC Scan

**API request**

curl -X GET "https://<QualysBaseURL>/cloudviewapi/rest/v1/iac/getScanList?filter=scanUuid:3010f375-084f-408f-9590-8a4692a5538c&pageNo=0&pageSize=80" -H 'authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

**Response**

```json
{
    "scanUuid": "3010f375-084f-408f-9590-8a4692a5538c",
    "scanDate": "2021-09-23T12:09:35.283+00:00",
    "name": "text file",
    "status": "ERROR",
    "tags": [],
    "message": "INTERNAL ERROR",
    "errorCode": 70503
}
```

Sample - Trigger an IaC Scan (SARIF)

**API request**

curl -X GET "https://<QualysBaseURL>/cloudviewapi/rest/v1/iac/scanResult?scanUuid=337a21ef-3c53-43bf-aed6-
Response (SARIF)

```json
{
    "version": "2.1.0",
    "runs": [
        {
            "tool": {
                "driver": {
                    "name": "QualysIaCSecurity",
                    "organization": "Qualys",
                    "rules": [
                        {
                            "id": "52140",
                            "name": "Ensure that Bucket should not log to itself",
                            "messageStrings": {
                                "remediation": {
                                    "text": "Ensure google_storage_bucket resource does not have argument log_bucket_name equal to bucket_name"
                                },
                                "criticality": {
                                    "text": "HIGH"
                                }
                            }
                        },
                        {
                            "id": "52036",
                            "name": "Ensure that Cloud Storage buckets have uniform bucket-level access enabled",
                            "messageStrings": {
                                "remediation": {
                                    "text": "Ensure google_storage_bucket resource has argument uniform_bucket_level_access set to True"
                                },
                                "criticality": {
                                    "text": "MEDIUM"
                                }
                            }
                        },
                        {
                            "id": "52030",
                            "name": "Ensure that Cloud Storage bucket is not anonymously or publicly accessible",
                            "messageStrings": {
                                "remediation": {
                                    "text": "Ensure google_storage_bucket resource does not have argument anonymous_access set to True"
                                },
                                "criticality": {
                                    "text": "HIGH"
                                }
                            }
                        }
                    ]
                }
            }
        }
    ]
}
```
"remediation": {
  "text": "Ensure google_storage_bucket_iam_member,
  google_storage_bucket_iam_binding resource does not have argument
  members set to allAuthenticatedUsers, allUsers"
},
"criticality": {
  "text": "HIGH"
}
],
"results": [
  {
    "ruleId": "52140",
    "level": "error",
    "message": {
      "text": "Ensure that Bucket should not log to itself"
    },
    "locations": [
      {
        "physicalLocation": {
          "artifactLocation": {
            "uri": "/GCPTF.tf"
          },
          "region": {
            "startLine": 1,
            "endLine": 11
          }
        }
      }
    ],
    "ruleId": "52036",
    "level": "error",
    "message": {
      "text": "Ensure that Cloud Storage buckets have uniform bucket-level
      access enabled"
    },
    "locations": [
      {
        "physicalLocation": {
          "artifactLocation": {
            "uri": "/GCPTF.tf"
          },
          "region": {
            "startLine": 1,
            "endLine": 11
          }
        }
      }
    ]
  }
]
{ "ruleId": "52030", "level": "error", "message": { "text": "Ensure that Cloud Storage bucket is not anonymously or publicly accessible" }, "locations": [ { "physicalLocation": { "artifactLocation": { "uri": "/GCPTF.tf" }, "region": { "startLine": 13, "endLine": 19 } } } ] }
Get IaC Scan List

/rest/v1/iac/getScanList

[GET]

You can fetch the list of scans that you have triggered by you. You could also use filters to narrow down the scan list. For example, filter such as status of the scan (SUBMITTED, PROCESSING, or FINISHED) to view scans that are in particular state.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter the scan list by providing a query using filters we support. The following search filters are supported:</td>
</tr>
<tr>
<td></td>
<td>- scanUuid: Unique identifier assigned to the IaC scan. The scan UUID is returned in the response of Trigger IaC scan after the scan is completed.</td>
</tr>
<tr>
<td></td>
<td>- status: status of the scan - SUBMITTED, PROCESSING, or FINISHED.</td>
</tr>
<tr>
<td></td>
<td>- tag.key &amp; tag.value: Use a text value # to define the key and value of the tag assigned to the resource (case sensitive). For example, using the status:FINISHED filter in the curl request fetches all scans that are completed.</td>
</tr>
<tr>
<td></td>
<td>- scanDate:[start date .. end date]: Use a date range or a specific date on which the scan was triggered. For more information on how to enter dates, see Date Queries.</td>
</tr>
<tr>
<td>pageNo</td>
<td>(integer) The page to be returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>(integer) The number of records per page to be included in the response.</td>
</tr>
</tbody>
</table>

Sample - Get Scan list
**API request**


**Response**

```json
{
    "content": [
        {
            "scanUuid": "337a21ef-3c53-43bf-aed6-46f04e1c542d",
            "tags": [
                {
                    "key": "Key",
                    "value": "Value"
                }
            ],
            "scanDate": "2021-06-16T12:05:03.889+00:00",
            "status": "FINISHED",
            "name": "FilterTrue"
        },
        ...
    ],
    "pageable": {
        "sort": {
            "sorted": true,
            "unsorted": false
        },
        "pageSize": 50,
        "pageNumber": 0,
        "offset": 0,
        "paged": true,
        "unpaged": false
    },
    "totalPages": 3,
    "totalElements": 140,
    "last": false,
    "number": 0,
    "size": 50,
    "numberOfElements": 50,
    "sort": {
        "sorted": true,
```
"unsorted": false,
"first": true
Understanding IaC Scan Output

The responses of IaC Scan APIs are in JSON format. In command line interface (CLI), the output is defaulted to tabular display. CLI can output JSON response with additional input parameter for format.

The response in JSON format has the following elements.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scanUuid</td>
<td>Unique identifier for the respective IaC scan</td>
</tr>
<tr>
<td>scanDate</td>
<td>Date when the scan was triggered.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the IaC scan.</td>
</tr>
<tr>
<td>status</td>
<td>Scan status. The values are: SUBMITTED, PROCESSING, or FINISHED.</td>
</tr>
<tr>
<td>tags</td>
<td>Tags from input.</td>
</tr>
<tr>
<td>result</td>
<td>This is a nested result which has details of findings. Refer next table for more details.</td>
</tr>
</tbody>
</table>

The "result" element has below sub-elements.

<table>
<thead>
<tr>
<th>Sub-Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkType</td>
<td>Type of check implemented on respective IaC templates. For example:terraform, terraform_plan, and so on.</td>
</tr>
<tr>
<td>results</td>
<td>Nested result structure which has details such as passed, failed, skipped checks, and parsing errors. Refer next table for further details.</td>
</tr>
<tr>
<td>summary</td>
<td>Summarizes count of passed, failed, skipped checks,</td>
</tr>
</tbody>
</table>
and parsing errors. For failed checks it shows stats about count of high, medium, log criticalities.

Each “results” element has below sub-elements.

<table>
<thead>
<tr>
<th>Sub-Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passedChecks</td>
<td>Lists all passed checks for IaC scan</td>
</tr>
<tr>
<td>failedChecks</td>
<td>Lists all failed checks for IaC scan</td>
</tr>
<tr>
<td>skippedChecks</td>
<td>Lists all skipped checks for IaC scan</td>
</tr>
<tr>
<td>parsingErrors</td>
<td>For the issues of parsing IaC templates, this section lists file names.</td>
</tr>
</tbody>
</table>

Each of passed, failed and skipped checks have evidence in response containing below fields.

<table>
<thead>
<tr>
<th>Sub-Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkId</td>
<td>An identifier of the check which was evaluated.</td>
</tr>
<tr>
<td>checkName</td>
<td>Description of check which was evaluated.</td>
</tr>
<tr>
<td>criticality</td>
<td>The criticality of the check and finding.</td>
</tr>
<tr>
<td>cvControl</td>
<td>Qualys CloudView control (run time). It has CID and Description</td>
</tr>
<tr>
<td>checkResult</td>
<td>It can either be PASSED, FAILED or SKIPPED. It also shows evaluated key for the respective check</td>
</tr>
<tr>
<td>codeBlock</td>
<td>The code block showing evidence in result.</td>
</tr>
<tr>
<td>filePath</td>
<td>Location of relative path of the template that was</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>fileLineRange</td>
<td>Line numbers impacted in respective IaC template.</td>
</tr>
<tr>
<td>repoFilePath</td>
<td>Location of relative path of the template that was scanned</td>
</tr>
<tr>
<td>resource</td>
<td>IaC template resource that was scanned.</td>
</tr>
<tr>
<td>callerFilePath</td>
<td>If terraform templates use modules, the evaluated code block is added. If called from other, sections are added to this field.</td>
</tr>
<tr>
<td>callerFileLineRange</td>
<td>If terraform templates use modules, the line numbers of caller are added to this field.</td>
</tr>
<tr>
<td>remediation</td>
<td>The remediation steps for customer's actions.</td>
</tr>
</tbody>
</table>