CloudView APIs

User Guide

Version 1.10.0

July 31, 2020
# Table of Contents

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

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

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

Azure APIs .................................................................................................................................. 32  

GCP APIs................................................................................................................................... 54  

Reports ..................................................................................................................................... 74  

Alerting Response APIs (Beta) ................................................................................................. 92  

Connector Groups Management APIs ...................................................................................... 165  

User Access Management APIs ............................................................................................... 172
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

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>
<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 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

```json
{
    "content": [
        {
            "name": "My_AWS_connector",
            "description": "string_change",
            "isGovCloud": false,
            "isChinaRegion": false,
            "awsAccountId": "XXXXXXXXXXXX",
            "state": "SUCCESS",
            "totalAssets": 4145,
            "groups": [
            
```
"name": "group_1",
"uuid": "bd8760b4-3fb9-38ef-be9b-7122924bcba3"
},
{
  "name": "grouP_2",
  "uuid": "eb9be4e9-3956-3b38-ad3f-9dbf7d5adbd3"
},
{
  "name": "group_3",
  "uuid": "86ec5b07-0a8d-3499-bb9f-43848960e43d"
}
"pollingFrequency": {
  "hours": 3,
  "minutes": 12
},
"provider": "AWS",
"connectorId": "1a1b6fb0-5150-11e9-b42c-7b6f80d86320",
"error": ",",
"baseAccountId": "XXXXXXXXXXXX",
"externalId": "1558121756190",
"arn": "arn:aws:iam::XXXXXXXXXXXX:role/CloudViewAWS",
"lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
"nextSyncedOn": "Fri Apr 1 13:45:08 UTC 2020",
"portalConnectorUuid": "81098071-d848-4e10-9fffd-efccace8312f",
"isPortalConnector": true
],
"pageable": {
  "sort": {
    "sorted": false,
    "unsorted": true
  },
  "pageSize": 50,
  "pageNumber": 0,
  "offset": 0,
  "paged": true,
  "unpaged": false
},
"totalElements": 1,
"last": true,
"totalPages": 1,
"first": true,
"sort": {
  "sorted": false,
  "unsorted": true
}
},
"numberOfElements": 1,
"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

```bash
curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/connectors/226947d0-569d-11e9-8032-2fa7ed9d9b64'
```

Response

```json
{
   "name": "My AWS Connector",
   "description": "string_check1",
   "isGovCloud": false,
   "isChinaRegion": false,
   "awsAccountId": "111111111111",
   "state": "SUCCESS",
   "totalAssets": 4234,
   "groups": [
      {
         "name": "group_1",
         "uuid": "bd8760b4-3fb9-38ef-be9b-7122924bcba3"
      },
      {
         "name": "group_2",
         "uuid": "eb9be4e9-3956-3b38-ad3f-9dbf7d5adbd3"
      }
   ]
}
```


```json
{
    "name": "group_3",
    "uuid": "86ec5b07-0a8d-3499-bb9f-43848960e43d"
},

"pollingFrequency": {
    "hours": 3,
    "minutes": 12
},

"provider": "AWS",
"connectorId": "226947d0-569d-11e9-8032-2fa7ed9d9b64",
"baseAccountId": "111111111111",
"externalId": "1532740312198",
"arn": "arn:aws:iam::111111111111:role/CloudViewRole",
"lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
"nextSyncedOn": "Fri Apr 1 13:45:08 UTC 2020",
"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": "XXXXXXXXXXXXXX",
  "chinaAccountId": "XXXXXXXXXXXXXXX",
  "govAccountId": "XXXXXXXXXXXXXXXXXXXXX",
  "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**

```json
{
    "content": [
        {
            "connectorName": "testts",
            "error": "Error getting config from af-south-1. Please check if region is enabled or config is enabled for this region",
            "occurredOn": "2020-07-09T12:41:50+0000",
            "region": null,
            "connectorId": "1e7fbcc0-89e7-11ea-a2c4-c9200d66f0b0"
        }
    ]
}
```
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",
  "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 external ID that you have
provided in AWS while creating the cross-account role.

- 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.

- 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::205767712438:role/QualysCloudView",
  "description": "string",
  "externalId": "1558121756190",
  "isChinaRegion": false,
  "isGovCloud": false,
  "isPortalConnector": true,
  "name": "My AWS Connector",
  "pollingFrequency": {
    "hours": 7,
    "minutes": 0
  }
}
```

**Response**

```
{
  "arn": "arn:aws:iam::XXXXXXXXXXXX:role/QualysCloudView",
  "awsAccountId": "XXXXXXXXXXXX",
  "baseAccountId": "XXXXXXXXXXXX",
  "connectorId": "1a1b6fb0-5150-11e9-b42c-7b8f80d86320",
  "description": "My AWS Connector",
  "error": "string",
```
"externalId": "1558121756190",
"groups": [],
"pollingFrequency": {
  "hours": 7,
  "minutes": 0
},
"isChinaRegion": false,
"isGovCloud": false,
"isPortalConnector": false,
"lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
"nextSyncedOn": "Fri Apr 1 17:33:08 UTC 2020",
"name": "My AWS Connector",
"portalConnectorUuid": "string",
"provider": "AWS",
"state": "PENDING",
"totalAssets": 0
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>
<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 - 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",
    "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
CloudView APIs

description stating the purpose of the connector you want to create.

-externalId: Specify the external ID that you have provided in AWS while creating the cross-account role.

-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.

- 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>/rest/v1/aws/connectors/{connectorId}'

Request POST Data

```
{
  "arn": "arn:aws:iam::XXXXXXXXXXXX:role/CloudViewARNrole",
  "description": "string_change",
  "externalId": "1558121756190",
  "isChinaRegion": false,
  "isGovCloud": false,
  "isPortalConnector": false,
  "name": "My updated AWS Connector",
  "pollingFrequency": {
    "hours": 4,
    "minutes": 0
  }
}
```

Response

```
{
  "arn": "arn:aws:iam::XXXXXXXXXXXX:role/QualysCloudViewRole",
  "awsAccountId": "XXXXXXXXXXXX",
  "baseAccountId": "XXXXXXXXXXXX",
  "connectorId": "1a1b6fb0-5150-11e9-b42c-7b8f80d86320",
  "description": "My updated AWS Connector",
  "error": "string",
  "externalId": "1558121756190",
  "groups": [],
  "pollingFrequency": {
    "hours": 4,
    "minutes": 0
  }
},
```
"lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
"nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020",
"isPortalConnector": false
"state": "PENDING",
"totalAssets": 0
}
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 - Get details of a specific connectors in user’s account

API request

curl -X DELETE -u <username>:<password> ' -d '77a51d30-14d4-12e9-aee4-31d950d53bd8' 'https://qualysapi.qualys.com/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}/{resourceId}/{connectorId}

[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>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

```
curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/aws/evaluations/stats/48/bizappsbucket/4589d050-3b21-11e9-9fd1-33f93691751d'
```

Response

```
{
"firstEvaluated": "2019-03-07T09:48:13+0000",
"lastEvaluated": "2019-03-26T08:17:06+0000",
"dateReopen": "2019-03-25T08:57:17+0000",
"dateFixed": "2019-03-25T08:51:45+0000"
}
```
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 7 days is included in the response. If you need data for specific date or date range, 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
Sample - Get the list of evaluations as per the account for AWS Controls

API request

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

Response

{
    "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
    },
    {
        "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
    },
    ...,
    {
        "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
    }
]
CloudView APIs

}

"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>
<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 7 days 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</td>
</tr>
</tbody>
</table>
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 all or filter the evaluations for your account**

**API request**


**Response**

```json
{
"content": [
{
"resourceId": "sample_resource",
"region": "us-east-1",
"accountId": "1111111111111",
"evaluatedOn": "2019-04-17T02:40:27+0000",
"evidences": [
{
"settingName": "Credential Report Generated Time",
"actualValue": "Tue Apr 16 00:24:25 UTC 2019"
},
{
"settingName": "MFA Status",
"actualValue": "Not Enabled"
},
{
"settingName": "Console Password Status",
"actualValue": "Enabled"
}
],
"resourceType": "IAM_USER",
"connectorId": "226947d0-569d-11e9-8032-2fa7ed9d9b64",
"result": "FAIL",
"evaluationDates":
{
```
"firstEvaluated": "2019-04-05T08:13:13+0000",
"lastEvaluated": "2019-04-17T02:40:27+0000",
"dateReopen": null,
"dateFixed": null
}
}
{
...
],
"pageable": {
"sort": {
  "unsorted": true,
  "sorted": false
},
"pageSize": 2,
"pageNumber": 0,
"offset": 0,
"paged": true,
"unpaged": false
},
"totalElements": 23,
"last": false,
"totalPages": 12,
"first": true,
"sort": {
  "unsorted": true,
  "sorted": false
},
"numberOfElements": 2,
"size": 2,
"number": 0
}
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
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>
<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 list of connectors in user’s account

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

API request


Response

```json
{
  "content": [
    {
      "name": "My Azure Connector",
      "description": "sample-description",
      "isGovCloud": true,
      "state": "REGIONS_DISCOVERED",
      "totalAssets": 244,
      "applicationId": "f076c321-694d-4929-ae0b-d2bd14d1a4d7",
      "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "groups": []
    }
  ]
}```
"provider": "AZURE",
"connectorId": "5926c280-c587-11e9-9230-31a5d0c73f76",
"lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
"directoryId": "ff4e2413-65ab-4dc2-9e5b-1ea02d3d94eb",
"pollingFrequency": {
  "hours": 4,
  "minutes": 0
},
"nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020",
},
{
  "name": "My Second Azure Connector",
  "description": "sample_description",
  "isGovCloud": true,
  "state": "SUCCESS",
  "totalAssets": 69,
  "applicationId": "d8c3a45a-e6f9-449e-8a54-2416e2d61aec",
  "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "groups": [
    {
      "name": "test_group",
      "uuid": "ce7c0d23-af60-320f-b8cb-f6b51b2fb8a5"
    },
    {
      "name": "sample_group",
      "uuid": "745aca99-6ab6-3e1d-8e65-7b4febc0005e"
    }
  ],
  "provider": "AZURE",
  "connectorId": "2e0c1660-d061-11e9-ad71-df4fba75b3c5",
  "lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
  "directoryId": "ff4e2413-65ab-4dc2-9e5b-1ea02d3d94eb",
  "pollingFrequency": {
    "hours": 4,
    "minutes": 0
  },
  "nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020",
}]
"pageable": {
  "sort": {
    "unsorted": true,
    "sorted": false
  },
  "pageSize": 50,
"pageNumber": 0,
"offset": 0,
"unpaged": false,
"paged": true
},
"totalElements": 2,
"last": true,
"totalPages": 1,
"first": true,
"sort": {
  "unsorted": true,
  "sorted": false
},
"numberOfElements": 2,
"size": 50,
"number": 0
}
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://qualysapi.qualys.com/cloudview-api/rest/v1/azure/connectors/2e0c1660-d061-11e9-ad71-df4fba75b3c5'

**Response**

```json
{
  "name": "My GovCloud Azure Connector",
  "description": "sample_description",
  "isGovCloud": true,
  "state": "SUCCESS",
  "totalAssets": 69,
  "applicationId": "d8c3a45a-e6f9-449e-8a54-2416e2d61ae6",
  "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "groups": [
    {
      "name": "sample_group",
      "uuid": "745aca99-6ab6-3e1d-8e65-7b4febc0005e"
    },
    {
      "name": "azure2",
```
"uuid": "ce7c0d23-af60-320f-b8cb-f6b51b2fb8a5"
  },
  "provider": "AZURE",
  "connectorId": "2e0c1660-d061-11e9-ad71-df4fba75b3c5",
  "lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
  "pollingFrequency": {
    "hours": 4,
    "minutes": 0
  },
  "nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020",
}
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>

```
{
    "applicationId": "string",
    "authenticationKey": "string",
    "description": "string",
    "directoryId": "string",
    "isGovCloud": false,
    "name": "string",
    "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 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.

-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 an Azure Connector

API request

```bash
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": "XXXXXXXXXXXXXXX",
   "description": "This is test description",
   "directoryId": "ff4e2442-65ab-4dc2-9e5b-1ea02d3d94eb",
   "isGovCloud": true,
   "name": "My Azure Connector",
   "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXX",
   "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",
   "state": "PENDING",
   "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXX",
   "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>
<tbody>
<tr>
<td>connectorRunRequest</td>
<td>(Array[string]) Specify the IDs of the connector that you want to execute/run. Example: [&quot;string&quot;]</td>
</tr>
</tbody>
</table>

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

Where,

-applciationId: 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.

- 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**

```bash
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",
    "pollingFrequency": {
        "hours": 6,
        "minutes": 0
    }
}
'https://<QualysURL>/cloudview-api/rest/v1/azure/connectors/7d80a840-40b6-11e9-9078-111111111111'
```

**Response**

```json
{
    "name": "My Azure Connector",
    "description": "Polling Frequency updated",
    "isGovCloud": false,
    "state": "PENDING",
    "totalAssets": 0,
    "applicationId": "d8c3a45a-e6f9-449e-8a54-2416e2d61aec",
    "subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "groups": [],
    "provider": "AZURE",
    "connectorId": "7d80a840-40b6-11e9-9078-111111111111",
    "lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
    "directoryId": "ff4e2413-65ab-4dc2-9e5b-1ea02d3d94eb",
    "pollingFrequency": {
        "hours": 6,
        "minutes": 0
    },
    "nextSyncedOn": "Fri Apr 1 16:33:08 UTC 2020"
}
```
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>
</tbody>
</table>
|             | Example:
|             | "67a51d30-14d4-12e9-aae4-31d950d53bd7"                      |

Sample - Delete AWS Connector

**API request**

```bash
curl -k -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
Azure Evaluations

Get the stats for specified control id and resource id

/rest/v1/azure/evaluations/stats/{controlId}/{resourceId}/{connectorId}

[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>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

curl -k -X GET -u <username>:<password> 'https://<QualysURL>/cloudview-api/rest/v1/azure/evaluations/stats/50035/assertion_USPOD02-vnet/742a20d0-4032-22e9-a677-053d7c82d8a3'

Response

```json
{
    "firstEvaluated": "2019-03-08T23:01:10+0000",
    "lastEvaluated": "2019-03-22T12:07:05+0000",
    "dateReopen": "2019-03-25T08:57:17+0000",
```
"dateFixed": "2019-03-25T08:51:45+0000"
Get the list of evaluations as per the account 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 7 days 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
Sample - Get all or filter the evaluations for your account

**API request**


**Response**

```json
{
    "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
        },
        ...
    ],
    "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
}```
CloudView APIs
Get the resources evaluated for the specified Azure account 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>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>
<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 all or filter the evaluations for your account**

**API request**

```bash
curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQK==' 
```

**Response**

```
{
  "content": [
  
```

```
"resourceId": "TestResource",
"subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX",
"evaluatedOn": "2019-03-22T11:06:07+0000",
"evidences": [
    {
        "settingName": "Data Encryption",
        "actualValue": "On",
        "expectedValue": ""
    }
],
"resourceType": "SQL_SERVER_DATABASE",
"result": "PASS",
"evaluationDates": {
    "firstEvaluated": "2019-03-13T15:02:07+0000",
    "lastEvaluated": "2019-03-22T11:06:07+0000",
    "dateReopen": null,
    "dateFixed": null
}
},
{
"resourceId": "dnd-automation-mcheck-pass",
"subscriptionId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXX",
"evaluatedOn": "2019-03-22T11:06:08+0000",
"evidences": [
    {
        "settingName": "Data Encryption",
        "actualValue": "On",
        "expectedValue": ""
    }
],
"resourceType": "SQL_SERVER_DATABASE",
"result": "PASS",
"evaluationDates": {
    "firstEvaluated": "2019-03-13T15:02:09+0000",
    "lastEvaluated": "2019-03-22T11:06:08+0000",
    "dateReopen": null,
    "dateFixed": null
}
],
"pageable": {
    "sort": {
        "unsorted": true,
        "sorted": false
    },
    "pageSize": 50,
"pageNumber": 0,
"offset": 0,
"unpaged": false,
"paged": true
},
"totalElements": 2,
"totalPages": 1,
"last": true,
"first": true,
"sort": {
  "unsorted": true,
  "sorted": false
},
"numberOfElements": 2,
"size": 50,
"number": 0
}
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
- 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>
<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 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 1",
      "description": "Test GCP Connector",
      "state": "SUCCESS",
      "totalAssets": 24,
      "projectId": "XXXXXXXX-XXXXX-XXXXX",
      "groups": [
        {
          "name": "GCPGroup",
          "uuid": "e1d1c379-0a79-3153-be99-05132a775e3d"
```
{    "name": "Sample_group",    "uuid": "86ec5b07-0a8d-3499-bb9f-43848960e43d" },    "name": "Example_group",    "uuid": "52660405-27d3-3f69-b764-b0061ab4c494" },    "name": "GCP_sample",    "uuid": "b3e9036d-b546-30d4-99fb-cb64b15efffa" }    "provider": "GCP",    "connectorId": "1111a111-1111-11a1-a1a1-1aa1a1111111",    "lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",    "pollingFrequency": {    "hours": 4,    "minutes": 0     },    "nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020" },    {    "name": "GCP Connector 2",    "description": "This is for test purposes",    "state": "SUCCESS",    "totalAssets": 115,    "projectId": "XXXXXXX-XXXXXX-XXXXXX",    "groups": [    {    "name": "Sample_group",    "uuid": "ea4b240f-c27c-30a6-ba28-8fc9a38fa8d1" },    {    "name": "GCP_group",    "uuid": "424042b9-16e8-3410-bfd1-86308d74638c" },    {    "name": "Sample_GCP",    "uuid": "6d515841-a02d-34d0-bab3-a83c4e18b3c3" }    ],    "provider": "GCP",    "connectorId": "1111b111-1111-11b1-b1b1-1bb1b1111111",    "lastSyncedOn": "Fri Apr 1 14:33:08 UTC 2020",}
"pollingFrequency": {
    "hours": 6,
    "minutes": 0
},

"nextSyncedOn": "Fri Apr 1 20:33:08 UTC 2020"
}

"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
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",
  "description": "Get connector details",
  "state": "SUCCESS",
  "totalAssets": 115,
  "projectId": "my-project-1524669048661",
  "groups": [
    {
      "name": "Sample_group",
      "uuid": "ea4b240f-c27c-30a6-ba28-8fc9a38fa8d1"
    },
    "provider": "GCP",
    "connectorId": "1111a111-1111-11a1-a1a1-1aa1a1111111",
    "lastSyncedOn": "Fri Apr 1 08:33:08 UTC 2020",
    "pollingFrequency": {
      "hours": 4,
```

```json
```
"minutes": 0,
"nextSyncedOn": "Fri Apr 1 12:33:08 UTC 2020"
Create GCP Connector

/rest/v1/gcp/connectors

[POST]

Specify the connector details such as name, description 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>
</tbody>
</table>

Sample - Create a new GCP Connector

API request

curl -k -X POST -d <username>:<password> 'Content-Type: multipart/form-data' --header 'Accept: application/json' -F name=My%20GCP%20Connector -F description=My%20GCP%20Connector -F "pollingFrequencyInHrs=4" -F "pollingFrequencyInMinutes=0" 'https://<QualysURL>/cloudview-api/rest/v1/gcp/connectors'

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

Response

```json
{
    "name": "My GCP Connector",
    "description": "My GCP Connector",
    "state": "PENDING",
    "totalAssets": 0,
    "projectId": "my-project-1111111111111",
    "provider": "GCP",
```
"connectorId": "1111a111-1111-11a1-a1a1-1aa1a1111111",
"lastSyncedOn": "Fri Apr 1 10:33:08 UTC 2020",
"pollingFrequency": {
  "hours": 4,
  "minutes": 0
},
"nextSyncedOn": "Fri Apr 1 14:33:08 UTC 2020"
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>[ &quot;string&quot;</td>
</tr>
<tr>
<td></td>
<td>]</td>
</tr>
</tbody>
</table>

Sample - Run the specified GCP connector

**API request**

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>configFile</td>
<td>Provide the configuration file.</td>
</tr>
</tbody>
</table>

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" '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",
"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"}
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 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}/{resourceId}/{connectorId}

[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>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

curl -k -X GET -u <username>:<password> 'Accept: application/json' -- 'https://<QualysURL>/cloudview-api/rest/v1/gcp/evaluations/stats/52019/3156296211597617506/1111a111-1111-11a1-a1a1-1aa1a1111111'

Response

```json
{
    "firstEvaluated": "2019-03-29T05:56:00+0000",
    "lastEvaluated": "2019-03-29T08:10:58+0000",
    "dateReopen": "2019-03-29T08:10:58+0000",
}
"dateFixed": "2019-03-29T06:09:05+0000"
Get the list of evaluations per account 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.</td>
</tr>
</tbody>
</table>

If you do not add a date filter, by default the data for last 7 days 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
### Sample - Get all or filter the evaluations for your account

<table>
<thead>
<tr>
<th><strong>API request</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>curl -X GET -u &lt;username&gt;:&lt;password&gt; 'https://&lt;QualysURL&gt;/cloudview-api/rest/v1/gcp/evaluations/my-project-11111111111111'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Response</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;content&quot;: [</td>
</tr>
<tr>
<td>{</td>
</tr>
</tbody>
</table>
|       "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 |
|     },
|     ... |
|     "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 account 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>
<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 7 days is included in the response. If you need data for specific date or date range, 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

<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-11111111111111111/resources/50010?pageNo=0&pageSize=50'
```

**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
      }
   ],
   "pageable": {
      "sort": {
         "sorted": false,
         "unsorted": true
      },
      "pageSize": 100,
      "pageNumber": 0,
      "offset": 0,
      "paged": true,
      "unpaged": false
   }
}
```


```json
{
  "totalElements": 1,
  "last": true,
  "totalPages": 1,
  "first": true,
  "sort": {
    "sorted": false,
    "unsorted": true
  },
  "numberOfElements": 1,
  "size": 100,
  "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

/\texttt{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

\texttt{curl -k -X GET -u \langle username\rangle: \langle password\rangle \ 'https://\langle Qualys\URL\rangle/\texttt{cloudview-api/rest/v1/reports/01164660-cfc7-11ea-a573-4395559d998e'}}

Response

```
{
    "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": "sample-azure-grp-1"
    }
  ],
  "cloudType": "GCP"
}
Get List of All Supported Mandates

/rest/v1/reports/mandates

[GET]

We support 25 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": 2762,
  "name": "NESA UAE Information Assurance Standards (IAS)"
},

{  
  "id": 2442,
  "name": "HITRUST Common Security Framework (CSF)"
},

{  
  "id": 2761,
  "name": "NIST 800-171 (Special Publication)"
},

{  
  "id": 3861,
  "name": "Sarbanes-Oxley Act: IT Security"
},

{  
  "id": 2605,
  "name": "Federal Risk and Authorization Management Program (FedRAMP M) - Moderate Security Baseline"
},

{  
  "id": 2803,
  "name": "General Data Protection Regulation (GDPR)"
IRDAI Guidelines On Information and Cyber Security for Insurers

Federal Risk and Authorization Management Program (FedRAMP H) - High Security Baseline
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'

**Response**

```json
[
  {
    "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"
  }
]```
{ "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**

```
{
  "content": [
    {
      "reportId": "84685b50-4d43-11e9-9496-d5e5ac80c6e2",
      "title": "Mandate_Report",
      "type": "MANDATE",
      "format": "ON SCREEN",
      "accounts": [
        {
          "name": "User_John",
          "id": "5a4f0630-39ab-11e9-a7c7-6f7103922bdf",
          "accountId": "11111111111",
          "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**

```
{
    "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:
{
    "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)

**groupIds**: 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",
  "groupIds": [
    "62edaf7d-4530-324c-be52-372a7acee33e"
  ],
  "mandateId": "2481",
  "policies": [
    {
      "cloudType": "GCP",
      "policyId": "10fffb910-3b6d-11ea-93a5-4d1356013529"
    }
  ]
}'
```
Response

{
   "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",
   "createdAt": "2020-07-27T05:07:01+0000",
   "cloudType": "GCP",
   "groupId": ["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. The syntax for the same is given below:</td>
</tr>
</tbody>
</table>

```
Syntax:
{
   "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)

**groupIds:** 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

<details>
<summary>API request</summary>

```bash
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",
  "groupIds": [
    "62edaf7d-4530-324c-be52-372a7acee33e"
  ],
  "mandateId": "2481",
  "policies": [
    {
      "cloudType": "GCP",
      "policyId": "10ffb910-3b6d-11ea-93a5-4d1356013529"
    }
  ]
}
```

</details>
Response
{
    "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": "10ffb910-3b6d-11ea-93a5-4d1356013529",
            "cloudType": "GCP"
        }
    ],
    "mandateId": "2481",
    "createdOn": "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
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

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>
<tbody>
<tr>
<td>filter</td>
<td>Form the search query using the filters we provide to refine the search for actions.</td>
</tr>
</tbody>
</table>

**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 |
CloudView APIs

<table>
<thead>
<tr>
<th>sortField</th>
<th>Specify the field that decides the sort order for the actions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sortOrder</td>
<td>Specify if the sorting needs to be ascending or descending order.</td>
</tr>
<tr>
<td>{asc</td>
<td>desc}</td>
</tr>
</tbody>
</table>

**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\n
  $\{control.criticality\}$ Severity Control Failure Detected for CID $\{cid\}\n  \n  *Affected Resource*
  
  \n  *Evaluation Summary*
  
  $\{evaluationDates\}$
  
  *Results*
  
  $\{control.result\}$
  
  $\{evidences\}$

```
```json
{
    "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",
    "subject": "Sample Pager Action",
    "pagerdutyServiceKey": "c391356a9d7d4c6b8a0257ff91cc3842",
    "pagerdutyEventType": "trigger",
    "activeRules": 0,
    "disabledRules": 0
}
```
Alerts

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

*Affected Resource*

\[
\begin{align*}
\text{resourceId:} & \text{${resource.id}} \\
\text{resourceType:} & \text{${resource.type}} \\
\text{service:} & \text{${service.type}} \\
\text{region:} & \text{${region}} \\
\text{cloudType:} & \text{${provider.type}} \\
\text{accountId:} & \text{${account.id}} \\
\text{connectorId:} & \text{${connectorUuid}} \\
\end{align*}
\]

*Evaluation Summary*

\[
\begin{align*}
\text{controlName:} & \text{${control.name}} \\
\text{controlId:} & \text{${cid}} \\
\text{policyName:} & \text{${policyName}} \\
\text{evaluatedOn:} & \text{${evaluatedOn}} \\
\text{evaluationDates:} & \text{${firstEvaluated}, ${lastEvaluated}} \\
\end{align*}
\]

*Results*

\[
\begin{align*}
\text{result:} & \text{${control.result}} \\
\text{evidences:} & \text{${evidences.key}}: \text{${evidences.value}} \\
\end{align*}
\]

Yours Sincerely,
Qualys Support Team

For any assistance, please contact our 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**

```bash
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}\n\n${resource.type}
```
CloudView APIs

---

*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

---
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"
   ]
}
```
<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Content</td>
</tr>
<tr>
<td>Response Code: 200</td>
</tr>
</tbody>
</table>
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"
}
```
CreatePagerDutyAction

/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>
<tbody>
<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>
<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;client&quot;: &quot;string&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;serviceKey&quot;: &quot;string&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;subjectLine&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>client:</td>
</tr>
<tr>
<td></td>
<td>serviceKey: the service key required to connect to your PagerDuty account.</td>
</tr>
<tr>
<td></td>
<td>subjectLine: subject of the action</td>
</tr>
</tbody>
</table>

**Sample - Create PagerDuty Action**

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

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

**Response**
```
{
  "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

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

#### Response

```json
{
    "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>
</tbody>
</table>

Example:

```
{
    "serviceKey": "c391356a9d7d4c6b8a0257ff91cc3123"
}
```

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>

```
{
  "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**
### API request

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

### Request PUT 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"
}
```
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>(body) Specify the channel and webhookUri of your Slack account.</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**

```
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/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"
}
```

**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

<table>
<thead>
<tr>
<th>API request</th>
</tr>
</thead>
<tbody>
<tr>
<td>curl -X GET --header 'Accept: application/json' --header 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQK==' 'https://&lt;QualysURL&gt;/cloudview-api/rest/v1/actions/types'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&quot;qemail&quot;, &quot;slack&quot;, &quot;pagerduty&quot;]</td>
</tr>
</tbody>
</table>
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).
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

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

Find actions that contain parts of the user name

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

Find actions that match exact value

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

**Examples**

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

**Examples**

Find actions updated by user ID

```
action.updatedById: jsmith
```

Find actions that contain parts of the user ID

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

Find actions that match exact value

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

**Examples**

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

**Examples**

Find action with 3 active rules

```
action.active : 3
```
CloudView APIs

Find action with more than 3 active rules

```
action.active > 3
```

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

**Examples**

Find action with 3 disabled rules

```
action.disabled : 3
```

Find action with more than 3 disabled rules

```
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

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

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

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

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

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

Show actions created on certain date

```
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

action.updatedDate: [2018-02-01 ... 2018-02-12]

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

action.updatedDate: [2018-02-01 ... now-1M]

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

action.updatedDate: [now-2w ... now-1s]

Show actions updated on certain date

action.updatedDate: '2018-02-22'

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
CloudView APIs

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`
```

**action.slackWebhookUri**

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

*Examples*

Find actions with Slack Webhook URI

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

Find actions that contain parts of the Slack Webhook URI

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

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

**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

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>
</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.

<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>
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 dXNlc3NhbWU6cGFzc3dvcmQK==' '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\n\n$control.criticality$ Severity Control Failure Detected for CID $cid$\n\nAffected Resource*: $resource.id$ $resource.type$ $service.type$ $region$ $cloudType$ $provider.type$ $account.id$ $connectorUuid$ $accountGroup$\n\n*Summary*: $control.name$ $policyName$ $evaluatedOn$ $firstEvaluated$ $lastEvaluated$ $result$ $evidences$\n\n*Settings*: $settings.name$ $settings.value$
```

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: $\{region\}$
ProviderType: $\{provider.type\}$
AccountId: $\{account.id\}$
TenantId: $\{tenant.id\}$
GroupId: $\{accountGroup\}$

Evaluation Summary

Control: $\{control.name\}$
ControlId: $\{controlId\}$

Policy Name: $\{policyName\}$
Evaluated On: $\{evaluatedOn\}$
Evaluation Dates:

First Evaluated: $\{firstEvaluated\}$
Last Evaluated: $\{lastEvaluated\}$

Results

Result: $\{control.result\}$
Evidences:

Name: $\{evidences.key\}$
Actual Value: $\{evidences.value\}$

Yours,
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 APIs"
null
CloudView APIs

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

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

Response

```json
{
  "actionId": "1f695df0-6da2-11ea-8910-77b847f40d61",
  "actionName": "Sample Slack Action",
  "actionType": "slack",
  "aggregate": true,
  "alert": "Qualys CloudView: Cloud Security Assessment Alerts

  MEDIUM Severity Control Failure Detected for CID 99*

  Affected Resource

  resourceId:arn:aws:lambda:us-east-1:205767712438:function:getEntitlementLambdaEast
  resourceType:LAMBDA
  service:LAMBDA
  region:us-east-1
tcloudType:AWS
taccountId:205767712438
tconnectedId:f8c3b440-4eaf-11ea-bbef-0dd8ca3bcd2e

  Evaluation Summary

  *tcontrolName:Ensure that Multiple Triggers are not configured in $Latest Lambda
```
Function
	controlId:99
	policyName:AWS Lambda Best Practices

tevaluatedOn:1585314208935

tevaluationDates:
		firstEvaluated:1581636116942
		lastEvaluated:1585314208935

*Results*
	result:FAIL
	evidences:
		settingName:[Total Triggers , Multiple
Triggers, Function Arn, Function Name, Role Arn]
		actualValue: [2, Yes, arn:aws:lambda:us-east-
1:205767712438:function:getEntitlementLambdaEast,
getEntitlementLambdaEast,
arn:aws:iam::205767712438:role/BizApps_Lambda_Role]

Yours
Sincerely,
Qualys Support Team

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

\[ \text{status: SUCCESS} \]

**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

\[ \text{statusDate: [2018-02-01 … 2018-02-12]} \]

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

\[ \text{statusDate: [2018-02-01 … now-1M]} \]

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

\[ \text{statusDate: [now-2w … now-1s]} \]

Show rule status modified on certain date

\[ \text{statusDate:'2018-02-22'} \]

**aggregate**

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

*Example*

Show aggregated rules

\[ \text{aggregate: TRUE} \]

**createdBy**

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

*Examples*

Find rules created by user

\[ \text{createdBy: Joe Smith} \]
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`

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`
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
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](#):
**Action Filters.**

- **pageNo** (integer) The page to be returned.
- **pageSize** (integer) The number of records per page to be included in the response.
- **sortField** Specify the field that decides the sort order for the rules.
- **sortOrder** \{asc|desc\} Specify if the sorting needs to be ascending or descending order.

**Sample - Get the list of rules**

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?cloudType=AWS&pageNo=1&pageSize=50&ruleType=simple_alert&sortOrder=asc'
```

**Response**

```
[
  {
    "id": "3dfc5050-7028-11ea-beeb-3fad76b6f6b5",
    "cloudType": "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}
	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}

tactualValue: ${evidences.value}

Yours Sincerely,
Qualys Support Team

For any assistance, please contact our support@qualys.com | 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

Alerts

Severity Control Failure Detected for CID ${cid}

Affected Resource

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

For any assistance, please contact our customer support team.

Sincerely,

Qualys Support Team

For any assistance, please contact our customer support team.

"emailRecipients": [
- abc@example.com
],

"slackChannel": null,

"subjectParameters": [],

"bodyParameters": []
]

"created": "2020-03-27T12:55:19.456+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
}
{ "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\n\n${control.criticality} Severity Control Failure Detected for CID ${cid}\n\n*Risk Group*\n	resourceId:${resource.id}\n	resourceType:${resource.type}\n	service:${service.type}\n	region:${region}\n	cloudType:${provider.type}\n	account:${account.id}\n	connector:connector\n
tgName:${accountGroup}\n\n*Evaluation Summary*\n
trolName:${control.name}\n	controlId:${cid}\n	不说:${policyName}\n	EvaluatedOn:${evaluatedOn}\n	EvaluationDates:[$(firstEvaluated) - $(lastEvaluated)\n\n*Results*\n	result:${control.result}\n	evidences:[$(evidences.key) = $(evidences.value)\n\n*Yours Sincerely, Qualys Support Team*\n\nFor any assistance, please contact our customer support team.", "emailRecipients": [ "abc@example.com" ], "slackChannel": null, "subjectParameters": [], "bodyParameters": [] }, ], "created": "2020-03-27T12:40:00.672+0000", "createdBy": "John Doe", "createdById": "user_john", "updated": "2020-03-27T12:40:00.672+0000", "updatedBy": "John Doe", "updatedById": "user_john", "lastRun": "2020-03-27T13:04:03.135+0000", "ruleState": "DISABLED", "durationHour": 0,
"fromHourInUTC": 0,
"fromMinuteInUTC": 0
},
{
"id": "dcf05f80-8ad1-11ea-9f4c-35b43d39dafc",
"cloudType": "AWS",
"ruleType": "simple_alert",
"name": "slack New template rule 01",
"description": "slack New Template",
"qql": "cid:99 and account.id:XXXXXXXXXXXXand
ccontrol.result:FAIL",
"aggregate": false,
"actions": [
{
  "id": "51cba540-8ad1-11ea-9f4c-35b43d39dafc",
  "actionType": "slack",
  "name": "slack new template",
  "subject": null,
  "alert": "Qualys CloudView: Cloud Security Assessment
Alerts\nAn assessment failure has been identified for resource
"${resource.id}" and control "${cid}" in your Qualys
subscription.\n*Impacted
Resource*\nresourceId:${resource.id}\nresourceType:${resource.type}

tservice:${service.type}\ntregion:${region}\ntcloudType:${provider.type}\ntaccountId:${account.id}\ntconnectorId:${connectorUuid}\ntgroupNamename:${accountGroup}\n
*Evaluation
Summary*\ntcontrolName:${control.name}\ntcontrolId:${cid}\ntpolicyName:${policyName}\nevaluatedOn:${evaluatedOnDateFormat}\ntevaluationDates:\nfirstEvaluated:${firstEvaluatedDateFormat}\nlastEvaluated:${lastEvaluatedDateFormat}\nEvidence*\nresult:${control.result}\nevidences:\nsettingName:${evidences.key}\ntactualValue:${evidences.value}\n
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": "2020-04-30T11:36.749+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**

```
{
  "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

${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}

taccountId:${account.id}
tconnectorId:${connectorUuid}
tgroupType:${accountGroup}

*Evaluation Summary*
	controlName:${control.name}
	policyName:${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": [
  "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.

```plaintext
aggregate
aggregationKey
durationHour
fromHourInUTC
fromMinuteInUTC
```

time_window_schedule_alert: For time_window_schedule_alert rule type, you need to provide all the parameters.
ruleBody (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
CloudView APIs

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

```json
{
    "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

```json
{
    "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:

e-mailSubject: 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. Example:</td>
</tr>
</tbody>
</table>

```json
{
   "ids": [
      "string"
   ]
}
```

Sample - Delete rules

API request

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

Request POST Data

```json
{
   "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.

Response
{
  "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>
</tbody>
</table>

```json
{
  "ids": [
    "string"
  ]
}
```

Sample - Disable rules

API request

```bash
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>
</tr>
</thead>
<tbody>
<tr>
<td>No Content</td>
</tr>
<tr>
<td>Response Code: 200</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>
</table>
| ruleIds   | (Array [string]) Specify the ID of an rule to be enabled and the rule gets enabled. You can provide multiple Ids separated by comma.  

Example:

```
{
    "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**

```
{
    "ids": [
    "1a841990-5dff-11ea-a923-6b29e6c4cbec",
    "efbf4080-52dd-11ea-a008-cbe911ab6a51"
    ]
}
```

**Response**
| 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

```text
ruleName: my first rule
```

Find rules that contain parts of the name

```text
ruleName: "my first rule"
```

Find rules that match exact value

```text
ruleName: `my first rule`
```

**ruleDescription**

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

*Examples*

Find rules with description

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

Find rules that contain parts of the description

```text
ruleDescription: "this rule is used for alerting"
```

Find rules that match exact value

```text
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

ruleQuery

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

```plaintext
aggregate: TRUE
```

**aggregationGroup**

Use *quotes* or *backticks* within values to find rules aggregated into a certain group.

**Examples**

Find rules with aggregation group

```plaintext
aggregationGroup: hostname
```

Find rules that contain parts of the aggregation group name

```plaintext
aggregationGroup: "hostname"
```

Find rules that match exact value

```plaintext
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

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

Find rules that contain parts of the message

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

Find rules that match exact value

```plaintext
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

action.subject: warning

Find rules that contain parts of the subject

action.subject: "warning"

Find rules that match exact value

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

action.slackChannel: Sec Ops

Find rules that contain parts of the slack channel name

action.slackChannel: "Sec Ops"

Find rules that match exact value

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`
Connector Groups Management APIs

Get Groups

/rest/v1/aws/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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</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 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

```json
{
  "content": [
    {
      "name": "empty",
      "uuid": "bad111c0-5de8-31ef-1111-4f1c2bd11111",
      "connectors": []
    },
    {
      "name": "sample_1",
      "uuid": "01011f80-fe1d-38c5-1111-56ac16361111",
      "connectors": [
        
      ]
    }
  ]
}
```
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**

```json
{
   "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>
</table>
| connectorGroupModifyRequest   | (body) Use this to specify the connector group IDs that you want to add and remove groups. 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: 111111111111)
  - Azure: subscription ID (Example: 11111111-1111-1111-1111-111111111111)
- GCP: project ID (Example: sample_gcp)

groupIds: Unique Id associated with each group.

Example: ea4b240f-c27c-30a6-ba28-8fc9a38fa8d1.

cloudType
Select the cloud provider to which the connector being updated belongs.

Sample - Update Connectors in a group

API request

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

Request POST Data

```
{
    "accountIdentifiers": [ "gcp-demo" ],
    "add":
    {
        "groupIds": [ "ea4b240f-c27c-30a6-ba28-8fc9a38fa8d1" ]
    },
    "remove":
    {
        "groupIds": [ "8b23977c-9f28-3007-8c11-c0469494053f" ]
    }
}
'https://<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
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",
      "connectorCount": 2
    }
  ]
}
```
"connectorName": "AWS_Connector_1"
],
"regions": [
    "us-east-1",
    "us-east-2"
],
"AZURE": {
    "directAccountScope": [
    {
        "connectorUuid": "2e0c1660-d061-11e9-ad71-df4fba75b3c5",
        "accountIdentifier": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXXXX",
        "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,

`groupId`: unique ID assigned to the group.

Example: ea4b240f-c27c-30a6-ba28-8fc9a38fa8d1

Sample - Update Connectors in a group

API request

```
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

```
{
   "add": {
      "accountIdentifiers": [
         "XXXXXXXXXXXX"
      ],
      "regions" : ["us-east-1","us-east-2"]
   },
   "remove": {
      "accountIdentifiers": ["XXXXXXXXXXXX"],
      "regions" : ["eu-west-1"]
   }
}
```

CloudType : AWS

Response

```
{
   "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"
    }
  ]
}