



Citrix Monitor Service API

Contents

About Citrix Monitor Service API	2
Data Access Protocol	8
Data Access methods	9
Securing On-Premises Monitor OData API Access	19
Descriptions for enums and error codes	20
Best Practices for performance optimization	38
Best Practices for performance optimization	39
How to invoke API	40
Machine data request examples	45
Machine data request examples	59
Machine resource utilization request examples	73
Session data request examples	80
Connection data request examples	92
Connection Failure log request examples	96
Application data request examples	101
Application instance data request examples	105
User data request examples	109
Cost optimization data request examples	114
Resources	117
Release notes	118

About Citrix Monitor Service API

January 14, 2026

Use the Citrix Monitor Service REST APIs to collect, analyze, and categorize data from the Citrix Virtual Apps and Desktops and Citrix DaaS (formerly Citrix Virtual Apps and Desktops service) site.

The Citrix Monitor Service API is built on the SQL Server Monitor database that is populated during processing and consolidation. This API is based on the ASP.NET Web API Framework. The Citrix Monitor Service API uses the [Open Data \(OData\) protocol](#), which is a Web protocol for querying and updating data, built upon Web technologies such as HTTP. ASP.NET Web API Framework supports OData Version 4.

You can access Citrix Monitor Service API using an OData consumer API. OData consumers are applications that consume data exposed using the OData protocol. For more access methods, please read [Data Access Methods](#).

If you are a Citrix Virtual Apps and Desktop administrator with read-only privileges, you can use the API to get following types of data from the Citrix Virtual Apps and Desktops and Citrix DaaS site:

- Data relating to connection failures, session usage, logon duration, and load balancing
- Machines in a failure state, hotfixes applied to a machine, hosted application usage, and application failure
- Machine level CPU, memory, and disk usage
- Process level CPU and memory usage

The preceding data received using the Citrix Monitor Service REST APIs allows you to:

- Analyze historical trends for future planning
- Perform detailed troubleshooting of connection and machine failures
- Extract information for feeding into other tools and processes; for example, using Microsoft Excel's PowerPivot tables to display the data in different ways
- Build a custom user interface based on the data provided by the API
- Run aggregation queries with the OData Version 4 endpoints to get basic grouping and aggregation functionality.

To get additional information on performing the preceding the tasks, go through the details on how to access the monitor service data, data access privilege, Data Access protocol.

Use the various access methods to export the data using OData API. Follow the Citrix Monitor Service REST APIs reference to get information on the methods, parameters, field descriptions, error types, and error values associated with the APIs.

Have a question on Citrix Monitor Service API? Find answers and ask questions, on the [Citrix Developer Exchange forum](#).

Supported API Gateway endpoints

Use one of the following endpoints based on the geographical region you selected while accessing the Citrix Monitor Service data for DaaS:

- US/EU/AP-S region: <https://api.cloud.com/monitorodata>
- Japan region: <https://api.citrixcloud.jp/monitorodata>
- US Gov region: <https://api.cloud.us/monitorodata>

Who can use this API service

To use the Monitor Service OData API, you must be a XenApp and XenDesktop administrator. To call the API, you require read-only privileges; however, the data returned is determined by XenApp and XenDesktop administrator roles and permissions.

For example, Delivery Group Administrators can call the Monitor Service API but the data they can obtain is controlled by Delivery Group access set up using Citrix Studio.

For more information about XenApp and XenDesktop administrator roles and permissions, see [Delegated Administration](#).

Data Available using the API

Currently, the following API entities are supported:

- **ApplicationActivitySummaries:** The activity summary of each application.
- **ApplicationErrors:** Records of application errors.
- **ApplicationFaults:** Records of application fault.
- **ApplicationInstances:** Record of an instance of a running application.
- **Applications:** Represents a virtualized application running on a session.
- **Catalogs:** Catalog images in the Site.
- **ConnectionFailureCategories:** Grouping for connection failure types.
- **ConnectionFailureLogs:** Log of each connection failure in the Site.
- **Connections:** Represents an initial connection or reconnect for a session.
- **DesktopGroups:** Delivery Groups in the Site.

- **DesktopOSDesktopSummaries:** Summary of Desktop usage.
- **FailureLogSummaries:** Failures (connection/machine) counts by time period and Delivery Group.
- **Hotfixes:** Hotfixes applied to a machine.
- **Hypervisors:** Hosts (hypervisors) in the Site.
- **LhcReport:** LhcReport.
- **LoadIndexSummaries:** Load Index averages by time period and machine.
- **LogOnSummaries:** Logon breakdown rollup/summary per hour/day.
- **LoadIndexes:** Load Index data received from the Virtual Delivery Agent (VDA).
- **LogOnMetrics:** Timestamps related to Interactive Session breakdown.
- **MachineCostSavingsSummaries:** Summary of machine cost saving metrics.
- **MachineCosts:** Model entity of the cost of a VM.
- **MachineFailureLogs:** Log of each machine failure by start and end date in the Site.
- **MachineHotfixLogs:** HotfixHistory relates the machine and hotfixes installed/ removed.
- **MachineMetric:** Represents a machine metrics statistics of a particular machine.
- **MachineMetricSummary:** Represents a machine metrics statistics summary of a particular machine.
- **MachineSummaries:** Summary of machine metrics.
- **Machines:** Machines in the Site.
- **NotificationActionEmails:** NotificationActionEmails.
- **NotificationEmailAddresses:** NotificationEmailAddresses.
- **NotificationEmailServerConfigurations:** NotificationEmailServerConfigurations.
- **NotificationRuleActions:** NotificationRuleActions.
- **NotificationRuleGroups:** NotificationRuleGroups.
- **NotificationRuleParameterChangeLogs:** NotificationRuleParameterChangeLogs.
- **NotificationRuleParameters:** NotificationRuleParameters.
- **NotificationRuleTargetValues:** NotificationRuleTargetValues.
- **NotificationRuleTargets:** NotificationRuleTargets.
- **NotificationRules:** NotificationRules.
- **NotificationSnmpConfigurations:** NotificationSnmpConfigurations.

- **NotificationTargets:** NotificationTargets.
- **Notifications:** Notifications.
- **NotificationsLogs:** NotificationsLogs.
- **ProbeDesktop:** Probe Desktop.
- **ProbeEndpoints:** Machines running Citrix Probe Agent to probe applications in the Site.
- **ProbeLogDesktop:** Probe Log Desktop.
- **ProbeLogs:** Results of probes run per application.
- **ProbeResultDesktop:** Probe Result Desktop.
- **ProbeResults:** Results of probes run (with failure stage) per application.
- **ProbeRuleApplications:** Probe Rule Application.
- **ProbeRuleDesktop:** Probe Rule Desktop.
- **ProbeRuleEndpoints:** Probe Rule Target.
- **ProbeRules:** Application probes configured in the Site.
- **ProbeStages:** Probe Stage.
- **ProcessDataLimitedMachines:** List the process data limited machines.
- **ProcessUtilization:** Represents a process utilization statistics of a particular process on a machine.
- **ProcessUtilizationDaySummary:** Day level summary for ProcessUtilization.
- **ProcessUtilizationHourSummary:** Hour level summary for ProcessUtilization.
- **ProcessUtilizationMinuteSummary:** Minute level summary for ProcessUtilization.
- **Processes:** Represents a process utilization statistics of a particular process on a machine.
- **PublishedDesktopNames:** Published Desktop Names.
- **ResourceUtilization:** Represents a resource utilization statistics of a particular machine every 5 minutes.
- **ResourceUtilizationSummary:** Represents a hourly level and day level resource utilization statistics of a particular machine.
- **ServerOSDesktopSummaries:** ServerOSDesktopSummaries.
- **SessionActivitySummaries:** Session counts and logon data by time period and delivery group.
- **SessionAutoReconnects:** Records of session auto reconnect.
- **SessionMetrics:** Represents a session metrics of a particular session for a machine.

- **SessionRecordingServers:** Session Recording Servers.
- **Sessions:** Represents a user connected to a desktop.
- **SpaApplications:** Represents a virtualized application running on a session for SPA service.
- **SpaEnumerations:** Represents the enumeration for SPA service.
- **SpaSessions:** Represents a user connected to a desktop for SPA service.
- **TaskLogs:** Log of all tasks and their status that have been run as part of the internal Monitor Service.
- **Users:** Users that have launched a session in the Site.

Please see [API Exploration](#) or [Schema Diagram](#) for schemas.

Authentication

Requests to the Monitor Service REST APIs require authentication. For **cloud** setups, the following are required for the request authentication:

- **Citrix CustomerID:** CustomerId is passed in the `Citrix-CustomerId` header
- **CCAuth Token:** Secure HTTP requests sent to Citrix Cloud (CC) use a custom `Authorization` header that contains a CC bearer token. Bearer tokens are required for actions that take place on behalf of a user.

For information on how to obtain the Citrix-CustomerId and generate a custom Authorization header that contains the bearer token, look at sections

[Get started with Citrix Cloud APIs](#) and [Generate Citrix Cloud bearer token](#).

The Monitor OData API is also available on **on-premises** setups.

The request URL is of the form: `http[s]://{ MonitorServiceHost } /Citrix/Monitor /OData/v4/Data/{ Entity }`, followed by the OData query parameters.

NTLM is used for authentication, so `username` and `password` need to be provided. The `username` may need to be prefixed with the `domain`, if applicable.

Required headers

Please set the following headers for the API requests for **cloud** setups

- **Citrix-CustomerId:** Your customer Id.
- **Authorization:** In format `CWSAuth bearer=<Token>`, `<Token>` is obtained from the [Authentication](#) process.

- **Accept-Encoding:** Set this header to `gzip`, `deflate`, `br` to reduce the network traffic. Some tools, such as Postman, will set this value automatically.
- **User-Agent:** Set this header to `<existing script/tool version>` can help Citrix fine tune the api performance and better rate limiting for customers who are leveraging OData for either internal automation and workflows or third party tools for data extraction, and monitoring.

Here is a sample request:

```
1 GET /monitorodata/$metadata HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
5 Accept-Encoding: gzip, deflate, br
6 User-Agent: third-party-tool-v1
```

Concurrent request limitation

For maintaining the service quality and preventing abuse, the service enforces a request concurrency limitation. If multiple concurrent requests are issued, new requests may fail with the “429 Too Many Requests” status code.

In such a scenario, fine-tune the script, increase the number of retries, extend the interval between queries, or simplify the query.

Pagination

Citrix Monitor supports OData pagination. All OData v4 endpoints return a maximum of 1000 records per page with a link to the next 1000 records. The following PowerShell script fetches applications in batches of 1000 records using Citrix Cloud authentication following the `@odata.nextLink` property in the response.

```
1 $customerId = "CUSTOMER_ID"
2 $apiEndpoint = "https://api.cloud.com/monitorodata"
3 $query = "$apiEndpoint/Applications"
4
5 $bearerToken = "CWSAuth bearer=TOKEN"
6 $headers = @{
7   'Citrix-CustomerId'=$customerId;'Authorization'=$bearerToken }
8
9
10 $results = Invoke-RestMethod $query -Headers $headers -Verbose
11 Write-Host "Number of items returned in the first call : ", $results.
    value.Count
12
13 while($results.'@odata.nextLink' -ne $null)
```

```
14 {
15
16     $results = Invoke-RestMethod $results.'@odata.nextLink' -Headers
17         $headers -Verbose
18     Write-Host "Number of items returned in next call : ", $results.
19         value.Count
20 }
```

In the above snippet, replace `CUSTOMER_ID` and `TOKEN` as needed for the cloud based [Authentication](#).

Data Access Protocol

July 29, 2024

The Monitor Service API is a REST-based API that can be accessed using an OData consumer. OData consumers are applications that consume data exposed using the OData protocol. OData consumers vary in sophistication from simple web browsers to custom applications that can take advantage of all the features of the OData Protocol.

Every part of the Monitor Service data model is accessible and can be filtered on the URL. OData provides a query language in the URL format you can use to retrieve entries from a service. You can use the OData V4 endpoints to run your queries. OData V4 supports aggregation queries.

The query is processed on the server side and can be filtered further using the OData protocol on the client side.

The data modeled falls into three categories: aggregate data (the summary tables), current state of objects (machines, sessions, etc.), and log data, which is really historical events (connections, for example).

Note: Enums are not supported in the OData protocol; integers are used in their place. To determine the values returned by the Monitor Service OData API.

What is OData Protocol?

[OData](#) (Open Data Protocol) is an [OASIS standard](#) that defines a set of best practices for building and consuming RESTful APIs. OData helps you focus on your business logic while building RESTful APIs without having to worry about the various approaches to define request and response headers, status codes, HTTP methods, URL conventions, media types, payload formats, query options, etc.

OData also provides guidance for tracking changes, defining functions/actions for reusable procedures, and sending asynchronous/batch requests.

OData RESTful APIs are easy to consume. The OData metadata, a machine-readable description of the data model of the APIs, enables the creation of powerful generic client proxies and tools.

OData Consumers: <http://www.odata.org/ecosystem/#consumers>

Client Libraries: <http://www.odata.org/libraries/>

Basic Tutorial: <http://www.odata.org/getting-started/basic-tutorial/>

As OData Provider, Citrix Monitor Service API is implemented by ASP.NET Web API.

For more detail about using ASP.NET Web API to create OData V4 endpoints, see [OData v4 Web API](#)

Data Access methods

June 2, 2025

The following examples show how to export Monitor Service data using the OData API.

Both **cloud** and **on-premises** setups are covered.

In the provided examples, authentication prerequisites are assumed to be available (e.g. credentials or bearer tokens). Please look at the [Authentication](#) section for details on the authentication methods.

Access using a Browser

1. Place the request URL into a web browser that is running with the appropriate administrative permissions for the XenApp and XenDesktop Site. Citrix recommends using the Chrome browser with a Rest Client add-in.
2. For on-premises setups, authenticate using a [username](#) and [password](#).
For cloud setups authenticate by appending the required headers to the request. See the [Authentication](#) paragraph for details.
3. View the response in a raw XML form.

Access using MS Excel

The steps provided below apply only to **on-premises** setups. For cloud setups, Power Query is required to implement the required Citrix Cloud authentication.

1. Click the Data tab.
2. Choose “Get Data”> “From Other Sources”> “From OData Feed”.

3. Enter the data feed URL: `http://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` (or `https:` if you are using SSL). Replace the placeholder values with the actual host and the OData query. Click **Next**.
4. An authentication window will popup during the first data load. Select **Windows** and provide the required credentials (`domain\username` and `password`). Click **Connect**.
5. Select the tables you want to import into Excel and click **Next**.
6. Transform the fetched data and finally Load it.

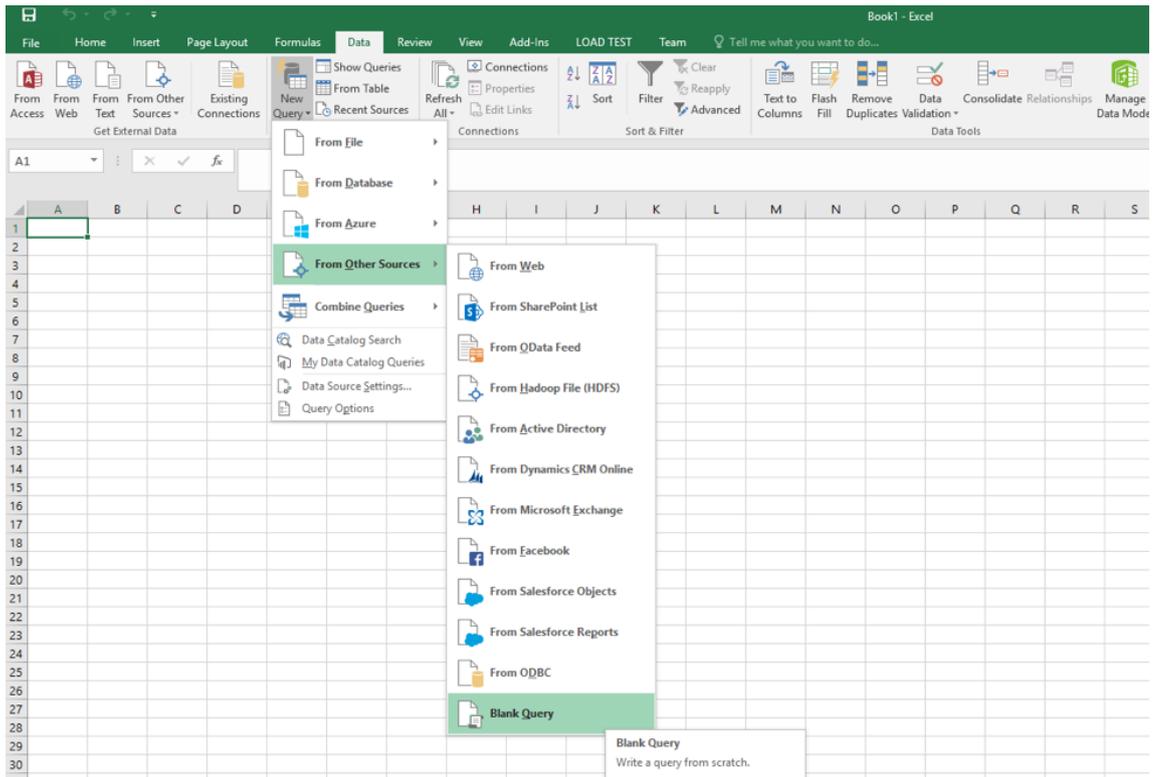
Access using MS Excel PowerQuery

[Power Query](#) is a data transformation and data preparation engine.

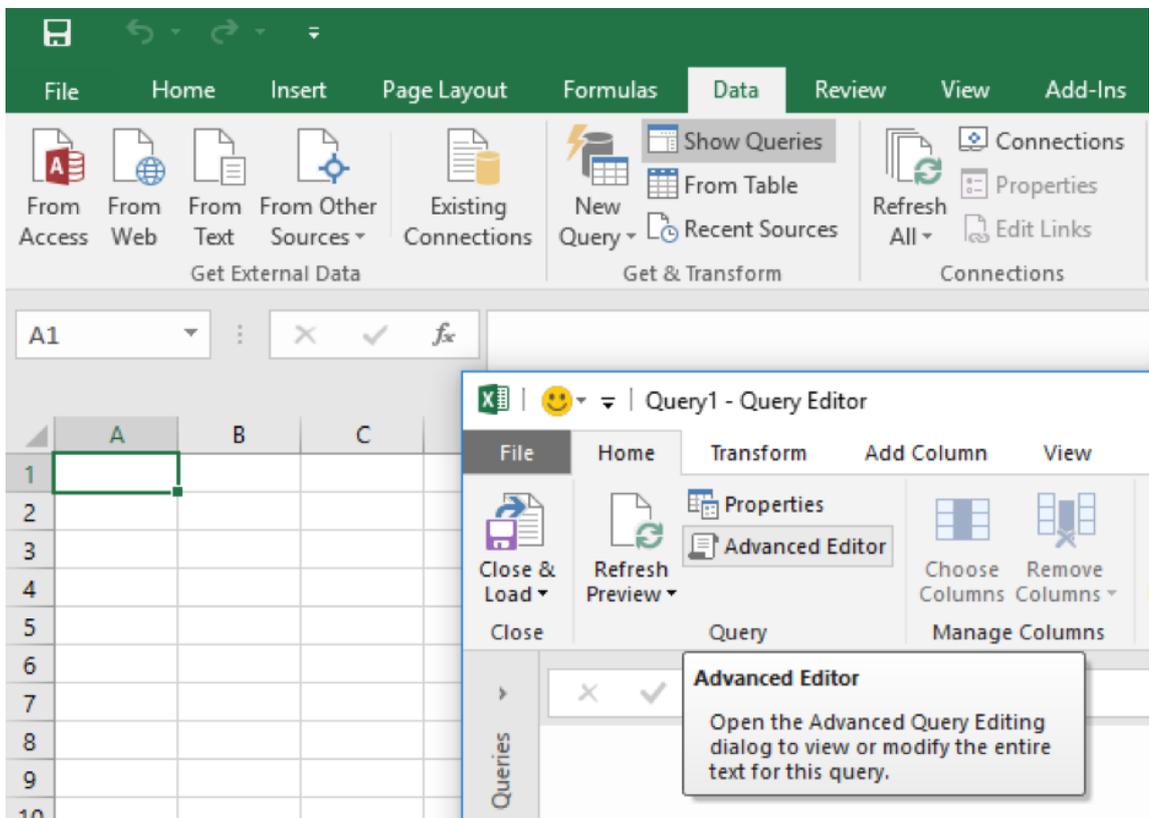
Power Query comes with a graphical interface for getting data from sources and a Power Query Editor for applying transformations. Because the engine is available in many [products and services](#), such as PowerBI and Excel.

Learn from the following example to access the Citrix Monitor Service data using MS Excel PowerQuery. This section only applies to **cloud** setups.

1. Open Excel (Excel 2016 has PowerQuery inbuilt. If you are using earlier versions of Excel, install PowerQuery. For more information, see [Microsoft](#) documentation).
2. In Excel 2016, go to **Data** menu > **Get & Transform group** > **New Query** > **From Other Sources** > click **Blank Query**



3. In the **Query Editor** window, go to **Home** menu > **Query group** > click **Advanced Editor**.



4. In the **Advanced Editor** window, type the following query:

```
“js
let
// Replace the place holder values with your customer and client values
customerId = “CUSTOMER_ID_PLACE_HOLDER”,
clientId = “CLIENT_ID_PLACE_HOLDER”,
clientSecret = “CLIENT_SECRET_PLACE_HOLDER”,
// Update the endpoint URL based on your region
ccAPIBaseURL = “https://api.cloud.com”,
// Replace the sample OData query
oDataQuery = “Machines?$filter=LifecycleState eq 0”,
```

```
1 requestURL = ccAPIBaseURL & "/monitorodata/" & oDataQuery,
2 tokenUrl = ccAPIBaseURL & "/cctrustoauth2/root/tokens/clients",
3
4 // Get Citrix Cloud API credential
5 headers = [
6     #"customerid" = customerId,
7     #"Content-Type" = "application/x-www-form-urlencoded",
8     #"Accept" = "*/*"
9 ],
10
11 postData = [
12     grant_type = "client_credentials",
13     client_id = clientId,
14     client_secret = clientSecret
15 ],
16
17 response = Json.Document(Web.Contents(tokenUrl, [Headers = headers
18     , Content = Text.ToBinary(Uri.BuildQueryString(postData))])),
19 token = "CwsAuth bearer=" & response[access_token],
20 apiHeaders = [
21     #"Authorization" = token,
22     #"Citrix-CustomerId" = customerId
23 ],
```

5. In case of first login, a message for specifying how to connect may be displayed. Click **Edit Credentials**. The **Access an OData feed** window appears. Make sure that the **Anonymous** tab and the base URL are selected and click **Connect**.
6. Upon successful authentication, the results are displayed as a table. In this example, the table lists all the Machines of the customer environment.

Access using LINQPad

This section refers to **on-premises** setups only.

1. Download and install the latest version of LINQPad from <http://www.linqpad.net>.
2. Run LINQPad with the appropriate administrative permissions for the XenApp and XenDesktop Site.

Tip: *the easiest way is to download, install and run on the Delivery Controller*

1. Click the Add connection link.
 - a. To use the OData v3 endpoint, choose WCF Data Services 5.1 (OData 3) and click **Next**.
 - b. To use the OData v4 endpoint the first time, click **View More Drivers**, search for the OData V4 Driver, click the **Download and Enable driver** link. This adds the Odata 4 driver to the list of available drivers. Subsequently, you can select OData 4 and click **Next**.
2. Enter the data feed URL: `http://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data` (or `https:` if you are using SSL). Select **Basic** authentication and provide the `username`, `password` and (optionally) `domain`. Click **OK**.
3. You can now run LINQ queries against the data feed and export the data as needed. For example, right-click Catalogs and choose **Catalogs.Take(100)**. This returns the first 100 Catalogs in the database. Choose Export>Export to Excel with formatting.

Access using a Client Library

Currently Citrix Monitor Service supports OData protocol V3 and V4. So, when implementing OData consumers with various programming platforms, please select the correct client libraries.

Access using C#/.NET

The code samples provided in this section demonstrate a way to access Citrix Monitor Service data using C# and more specifically the *Simple.OData.Client* package.

First, the *Simple.OData.V4.Client* Nuget Package needs to be installed. **Note:** If used along with *Microsoft.OData.Core*, the latter may need to be downgraded to a compatible version.

Sample code for **cloud** scenario:

```
1 using Simple.OData.Client;
2
3 namespace ODataClientExample
4 {
5
6     class Program
7     {
8
9         static async Task Main()
```

```
10     {
11
12         var settings = new ODataClientSettings();
13         settings.BaseUri = new Uri("https://api.cloud.com/
14             monitorodata/");
15         settings.BeforeRequest += request =>
16         {
17             request.Headers.Add("Authorization", "CWSAuth bearer=
18                 BEARER_TOKEN");
19             request.Headers.Add("Citrix-CustomerId", "CUSTOMER_ID")
20                 ;
21         }
22     ;
23     var client = new ODataClient(settings);
24     var x = ODataDynamic.Expression;
25     var machines = await client
26         .For(x.Machine)
27         .FindEntriesAsync();
28     foreach (var machine in machines)
29     {
30         Console.WriteLine($"MachineID: {
31             machine.Id }
32             , Name: {
33             machine.Name }
34             ");
35     }
36 }
37
38 }
39
40 }
```

The URI can be adjusted to match the customer region. The `CUSTOMER_ID` should be replaced with the Citrix Cloud Customer ID. Finally, the `BEARER_TOKEN` can be fetched dynamically using C# code samples available under page [Generate Citrix Cloud bearer token](#).

Sample code for **on-premises** scenario:

```
1 using Simple.OData.Client;
2 using System.Diagnostics;
3 using System.Net;
4
5 namespace ODataClientExample
6 {
7
8     class Program
9     {
10
11         static async Task Main()
12         {
```

```
13
14     var credentials = new NetworkCredential("DOMAIN\USERNAME",
15     "PASSWORD");
16     var baseUrl = new Uri("MONITOR_SERVICE_HOST/Citrix/Monitor/
17     OData/v4/Data/");
18     var settings = new ODataClientSettings(baseUrl, credentials
19     )
20     {
21         PayloadFormat = ODataPayloadFormat.Json,
22         IgnoreResourceNotFoundException = true,
23         IgnoreUnmappedProperties = true,
24         RenewHttpConnection = true
25     };
26     var client = new ODataClient(settings);
27     var x = ODataDynamic.Expression;
28     var machines = await client
29     .For(x.Machine)
30     .FindEntriesAsync();
31     foreach (var machine in machines)
32     {
33         Console.WriteLine($"Machine ID: {
34     machine.Id }
35     , Name: {
36     machine.Name }
37     ");
38     }
39
40     }
41
42     }
43
44 }
```

Replace the placeholder credentials (USERNAME and PASSWORD), as well as the MONITOR_SERVICE_HOST value.

References:

- [Getting started with Simple.OData.Client](#)
- [OData Tutorial with Simple.OData.Client](#)
- [Calling an OData Service From a .NET Client \(C#\)](#)

Access using Java**Calling an OData Service from a Java Client based on Odata4j v0.3 library:**

<http://www.odata.org/libraries/>

<http://odata4j.org/v/0.3/javadoc/>

Code Fragment:

```
1 // create consumer instance
2 String serviceUrl = "http://{
3   ApiGatewayEndpoint }
4   ";
5 ODataConsumer consumer = ODataConsumer.create(serviceUrl);
```

```
1 // =====General Query=====
2 Enumerable<String> qEntitySets = consumer.getEntitySets();
3 System.out.println(qEntitySets.first().toString());
4
5 Enumerable<OEntity> qList = consumer.getEntities(qEntitySets.first()).
   execute();
6 System.out.println(qList.first().toString());
7
8 OEntity qEntity = qList.first();
9 System.out.println(qEntity.getProperties().get(0));
10
11 OProperty<?> qProperty = qEntity.getProperties().get(0);
12 System.out.println(qProperty.getName());
13 System.out.println(qProperty.getType());
14 System.out.println(qProperty.getValue());
```

```
1 // =====Filter Query
   =====
2 /* GET http://{
3   ApiGatewayEndpoint }
4   /Machines */
5 String entitySetName = "Machines";
6 qList = consumer.getEntities(entitySetName).execute();
7 System.out.println(qList.first().toString());
8
9 /* GET http://{
10  ApiGatewayEndpoint }
11  /Machines()?$select=Name, IPAddress */
12 qList = consumer.getEntities(entitySetName).select("Name,IPAddress").
   execute();
13 System.out.println(qList.first().toString());
14
15 /* GET http://{
16  ApiGatewayEndpoint }
17  /Machines()?$orderby=Id desc&$skip=40&$top=10 */
18 qList = consumer.getEntities(entitySetName).orderBy("Id desc").skip(2).
   top(10).execute();
19 System.out.println(qList.first().toString());
20
21 /* GET http://{
22  ApiGatewayEndpoint }
23  /Machines()?$filter=Name eq '$Name'*/
24 qList = consumer.getEntities(entitySetName).filter("Name eq 'DOMAIN\\
```

```
    HOSTNAME'").execute();  
25 System.out.println(qList.first().toString());
```

Access using PowerShell

Learn from the following examples how to access the Citrix Monitor Service data using PowerShell.

Sample OData query for **cloud** scenario, triggered from PowerShell with the headers initialized and the Raw XML output redirected to a file:

```
1 $customerId = "CUSTOMER_ID"  
2 $bearerToken = "CWSAuth bearer=TOKEN"  
3 $apiEndpoint = "https://api.cloud.com/monitorodata"  
4 $query = "$apiEndpoint/Applications"  
5 $headers = @(  
6   'Citrix-CustomerId'=$customerId;'Authorization'=$bearerToken )  
7  
8 $results = Invoke-RestMethod $endpoint -Headers $headers -Verbose  
9 \$results | ConvertTo-Json > <Path-to-Output-File>
```

Please look at section [Generate Citrix Cloud bearer token](#) for a PowerShell example for generating a bearer token.

Same query, adjusted for the **on-premise** scenario:

```
1 $userName = 'domain\username'  
2 $userPassword = 'password'  
3 [securestring]$secStringPassword = ConvertTo-SecureString $userPassword  
   -AsPlainText -Force  
4 [pscredential]$credObject = New-Object System.Management.Automation.  
   PSCredential ($userName, $secStringPassword)  
5 $url = "http[s]://{  
6   MonitorServiceHost }  
7   /Citrix/Monitor/OData/v4/Data/User"  
8 $results = Invoke-WebRequest -Uri $url -Method GET -Credential  
   $credObject  
9 $results.Content > <Path-to-Output-File>
```

Complete solutions for collecting Monitor Service data using PowerShell scripts can be found under the [citrix-data](#) GitHub repository.

Access using any REST API tool

Learn from the following example how to access the Citrix Monitor Service data using any REST API tool, for example Postman Chrome Ext.

[Postman](#) is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration, so you can create better APIs, faster.

1. Install Postman from Chrome extensions. See <https://chrome.google.com/webstore>.
2. Launch the application.
3. Select GET in the dropdown and type in the URL to request.



1. Insert headers with your token and customerId. Click **Send**.
2. If authentication is successful, response data is seen in the **Response** section of Postman.

For accessing **on-premise** Monitor Service data, use NTLM authentication instead in the Authorization tab and provide **username** and **password**.

A Postman collection including several OData examples can be found at [Monitor Service OData Postman Collection](#).

Access using Python script

Learn from the following example to access the Citrix Monitor Service data using Python script with [Requests](#) library.

For a **cloud** environment:

```
1 import requests
2 url = "https://{
3   ApiGatewayEndpoint }
4   /monitorodata/Machines"
5 payload = {
6   }
7
8 headers = {
9
10  'Authorization': 'CWSAuth bearer=BEARER_TOKEN',
11  'Citrix-CustomerId': 'CUSTOMER_ID'
12 }
13
14 response = requests.request("GET", url, headers=headers, data=payload)
15 print(response.text)
```

Replace **CUSTOMER_ID** and **BEARER_TOKEN** with the values of CustomerID and bearer token. An example for generating a bearer token using Python can be found under section [Generate Citrix Cloud bearer token](#).

For an **on-premises** environment:

```
1 import requests
2 from requests_ntlm import HttpNtlmAuth
```

```
3 url = "http[s]://{
4   MonitorServiceHost }
5   /Citrix/Monitor/odata/v4/Data/Machines"
6 response = requests.get(url, auth=HttpNtlmAuth('domain\\username', '
7   password'))
8 print(response.text)
```

Access using curl

Accessing Citrix Monitor Service data is also feasible using command `curl`.

For a **cloud** environment:

```
1 curl --location 'https://{
2   ApiGatewayEndpoint }
3   /monitorodata/Machines' \
4     --header 'Citrix-CustomerId: {
5   YourCustomerId }
6   ' \
7     --header 'Authorization: CWSAuth bearer={
8   YourToken }
9   ' \
10    --header 'User-Agent: curl'
```

For an **on-premises** environment:

```
1 curl --ntlm --user 'domain\username:password' \
2     --location 'http[s]://{
3   MonitorServiceHost }
4   /Citrix/Monitor/odata/v4/Data/Machines'
```

Securing On-Premises Monitor OData API Access

April 8, 2025

IMPORTANT:

This section is NOT applicable for cloud customers. Citrix DaaS always requires HTTPS connections using TLS.

It is recommended that you configure the Monitor Service for TLS. This requires installing a suitable certificate on the Delivery Controller and configuring the Monitor Service. For more information, see [Enable TLS on Delivery Controllers](#). Once you have done this, always connect to the monitor endpoint over HTTPS. You must ensure that the client device trusts the certificate used on the Delivery Controller, and that the client uses TLS 1.2 or higher.

Descriptions for enums and error codes

January 15, 2026

AllocationType

Error code	Enum value	Description
0	Unknown	Unknown. When the Broker does not send an allocation type.
1	Static	Machines get assigned to a user either by the admin or on first use. This relationship is static and changes only if an admin explicitly changes the assignments.
2	Random	Machines are allocated to users randomly from a pool of available machines.
3	Permanent	Equivalent to 'Static'.

ConnectionFailureEnumValue

Error Code	Enum value	Description
0	Unknown	Unknown.
1	None	None (no failure).
2	SessionPreparation	Failure during session preparation.
3	RegistrationTimeout	Failure due to registration timeout.
4	ConnectionTimeout	Failure due to connection timeout.
5	Licensing	Failure due to licensing.
6	Ticketing	Failure due to ticketing.

Error Code	Enum value	Description
7	Other	Other failure reasons.
8	GeneralFail	General failure.
9	MaintenanceMode	Desktop group, machine, or hypervisor is in maintenance mode.
10	ApplicationDisabled	Selected application is currently disabled.
11	LicenseFeatureRefused	Required feature is not licensed.
12	NoDesktopAvailable	No machine is available to satisfy launch.
13	SessionLimitReached	VDI machine is already in use (not used for RDS).
14	DisallowedProtocol	Requested protocol is not allowed.
15	ResourceUnavailable	Resource is unavailable.
16	ActiveSessionReconnectDisabled	Active session stealing is required, but is disabled.
17	NoSessionToReconnect	Session to which reconnect is directed is not found (can only occur on a launch retry).
18	SpinUpFailed	Failed to power-up machine for launch.
19	Refused	Session refused.
20	ConfigurationSetFailure	Configuration set failure.
21	MaxTotalInstancesExceeded	App launch refused because limit on total concurrent usage is reached.
22	MaxPerUserInstancesExceeded	App launch refused because limit on per-user usage is reached.
23	CommunicationError	Launch failed because the VDA could not be contacted.
24	MaxPerMachineInstancesExceeded	App launch refused because limit on per machine usage is reached.

Error Code	Enum value	Description
25	MaxPerEntitlementInstancesExceeded	Desktop launch refused because limit on per entitlement usage is reached.
100	NoMachineAvailable	No machine available.
101	MachineNotFunctional	Machine not functional.

CatalogType

Error code	Enum value	Description
0	ThinCloned	A thin-cloned catalog is used for original golden VM images that are cloned when they are assigned to a VM, and users' changes to the VM image are retained after the VM is restarted.
1	SingleImage	A single-image catalog is used when multiple machines provisioned with Provisioning Services for VMs all share a single golden VM image when they run and, when restarted, they revert to the original VM image state.
2	PowerManaged	This catalog kind is for managed machines that are manually provisioned by administrators. All machines in this type of catalog are managed, and so must be associated with a hypervisor connection.

Error code	Enum value	Description
3	UnManaged	This catalog kind is for unmanaged machines, so there is no associated hypervisor connection.
4	Pvs	This catalog kind is for managed machines that are provisioned using Provisioning Services. All machines in this type of catalog are managed, and so must be associated with a hypervisor connection. Only shared desktops are suitable for this catalog kind.
5	Pvd	A personal vDisk catalog is similar to a single-image catalog, but it also uses personal vDisk technology.
6	PvsPvd	A Provisioning Services-personal vDisk (PvsPvd) catalog is similar to a Provisioning Services catalog, but it also uses personal vDisk technology.

ConditionTargetType

Error code	Enum value	Description
0	Unknown	The unknown.
1	Site	The site.
2	Controller	The controller.
3	DesktopGroup	The delivery group.
4	Catalog	The Catalog.
5	RdsWorker	The RDS worker.

Error code	Enum value	Description
6	Vdi	The vdi.
7	User	User.

ConnectionFailureType

Error code	Enum value	Description
0	None	None.
1	ClientConnectionFailure	Session failed to start.
2	MachineFailure	Machine failure.
3	NoCapacityAvailable	No available capacity to start or reconnect to a session.
4	NoLicensesAvailable	No Licenses available.
5	Configuration	Configuration failure.

ConnectionState

Error code	Enum value	Description
0	Unknown	Default value: unknown.
1	Connected	Actively connected to desktop.
2	Disconnected	Disconnected from desktop, but session still exists.
3	Terminated	Session has been terminated.
4	PreparingSession	Session is in the preparing state.
5	Active	Session is active.
6	Reconnecting	User is reconnecting to the session.
7	NonBrokeredSession	Session is non-brokered.
8	Other	Connection state is reported as Other.

Error code	Enum value	Description
9	Pending	Connection state is pending.

DeliveryType

Error code	Enum value	Description
0	DesktopsOnly	Only desktops are published.
1	AppsOnly	Only applications are published.
2	DesktopsAndApps	Both desktops and applications are published.

DeregistrationReasonCode

Error Code	Name	Description
0	AgentShutdown	VDA reported shutdown via CBP.
1	AgentSuspended	VDA reported OS suspended via CBP.
2	AgentRequested	Failure during SessionPreparation.
100	IncompatibleVersion	Broker agent version on the VDA is too old.
101	AgentAddressResolutionFailed	Broker cannot find VDA machine information in AD (the VDA machine SPN).
102	AgentNotContactable	Test callback from broker to VDA failed.
103	AgentWrongActiveDirectoryOU	When using AD-based discovery on the VDA, and the AD OU used by the VDA is the wrong one for this site.
104	EmptyRegistrationRequest	Registration request body from VDA is empty/missing.

Error Code	Name	Description
105	MissingRegistrationCapabilities	Expected callback endpoints from the registration request are missing.
106	MissingAgentVersion	Registration request does not specify agent version.
108	NotLicensedForFeature	Required edition-specific feature is unavailable for configured license.
109	UnsupportedCredentialSecurityVersion	Requested credential encryption method is not supported by the broker.
110	InvalidRegistrationRequest	Registration request contains details that are not allowed or corrupted.
111	SingleMultiSessionMismatch	VDA is configured to be in a catalog/group for single/multi session VDAs but reports as the opposite kind.
112	FunctionalLevelTooLowForCatalog	VDA reported functional level is lower than the catalog configured minimum functional level.
113	FunctionalLevelTooLowForDesktopGroup	VDA reported functional level is lower than the group configured minimum functional level.
114	OSNotCompatibleWithDdc	Some types of Windows OS, such as multi-session win10, are not allowed to use with on-premises Citrix Virtual Apps and Desktops.
115	VMNotCompatibleWithDdc	Some types of VM, such as multi-session win10 on non-Azure hosted machines, are not allowed.

Error Code	Name	Description
200	PowerOff	VDA is marked as deregistered because the VDA is reported as powered off.
201	DesktopRestart	End user requested a restart of the VDA.
202	DesktopRemoved	VDA has been removed from its delivery group.
203	AgentRejectedSettingsUpdate	CBP call to Configuration.Set failed because the VDA broker agent rejected it.
204	SendSettingsFailure	CBP call to Configuration.Set failed for some other reason.
205	SessionAuditFailure	CBP call to request a session audit failed.
206	SessionPrepareFailure	CBP call to prepare for a session launch failed.
207	ContactLost	VDA has not contacted the broker (eg with a ping) for longer than the timeout period.
208	SettingsCreationFailure	Settings could not be sent to the VDA because preparing the settings to send failed.
300	UnknownError	General failure, such as an unexpected exception.
301	BrokerRegistrationLimitReached	Broker cannot accept another VDA registration based on the configured limit.
400	None	For power action case, no deregistration.
401	HypervisorReportedFailure	For power action case, hypervisor request failed.
402	HypervisorRateLimitExceeded	For power action case, hypervisor request refused due to quotas.

Error Code	Name	Description
1000	HardRegistrationPending	Registration success. VDA hard registration is pending.
1001	SoftRegistered	Registration success. VDA is soft registered.
1002	Unknown	Registration failure. Internal reason code does not map to opevent code.

DesktopKind

Error code	Enum value	Description
0	Private	Private desktops.
1	Shared	Shared desktops.

DesktopType

Error code	Enum value	Description
0	None	None.
1	Vdi	VM hosted desktop.
2	RemotePc	Unmanaged desktop RemotePC.
3	Rds	VM hosted RDS.
4	Unknown	Unknown type.

FailureCategory

Error code	Enum value	Description
0	Unknown	Default value: unknown.
1	Connection	Connection failure category.

Error code	Enum value	Description
2	Machine	Machine failure category.

LifecycleState

Error code	Enum value	Description
0	Active	Default value: entity is active.
1	Deleted	Object was deleted.
2	RequiresResolution	Object was created, but values are missing, so a background process should poll to update missing values.
3	Stub	Stub object: for example, a machine or a session that does not exist but is created by internal processing logic to preserve data relationships.

LogonBreakdownType

Error code	Enum value	Description
0	None	Default value: none.
1	UsersLastSession	User's last session information.
2	UsersSessionAverage	Average across the user's sessions.
3	DesktopGroupAverage	Average for the desktop group.

LogOnStep

Error code	Enum value	Description
0	Total	Total.
1	Brokering	Time taken to complete the process of brokering the session.
2	VMStart	If the session required a machine to be started, the time taken to start the VM.
3	Hdx	Time taken to complete the steps required in setting up the HDX connection from the client to the VM.
4	Authentication	Time taken to complete authentication to the remote session.
5	Gpos	If any Group Policy settings have been enabled on the machines, the time taken for the GPOs to be applied.
6	LogOnScripts	If logon scripts are configured for the session, the time taken for the logon scripts to be executed.
7	ProfileLoad	If profile settings are configured for the user or the machine, the time taken for the profile to be loaded.
8	Interactive	Time taken to hand off keyboard and mouse control to the user.

MachineFaultStateCode

Error code	Enum value	Description
0	Unknown	Fault unknown.
1	None	No fault (healthy machine).

Error code	Enum value	Description
2	FailedToStart	Last power-on operation for the machine failed.
3	StuckOnBoot	Machine might not have booted following power on (VM tools did not transition to running).
4	Unregistered	Machine has failed to register within the expected period, or its registration has been rejected.
5	MaxCapacity	Machine is reporting itself at maximum capacity.
6	VirtualMachineNotFound	Virtual machine is not found.

MachineRole

Error code	Enum value	Description
0	Vda	Vda machine.
1	Ddc	Ddc machine.
2	Both	Machine acting as VDA and DDC.

PowerActionReasonCode

Error code	Enum value	Description
0	Unknown	Unknown reason.
1	Reset	For shutdown power actions when a user requests the reset of a VM from WI/Storefront.

Error code	Enum value	Description
2	Pvd	For 'turn on' power actions when a VM needs to have its associated Pvd disk updated due to a change in the underlying master OS disk image.
3	Schedule	For shutdown and turn on power actions to be generated to reboot machines at configured times and/or to reboot machines to roll out updated base master OS image disks to VMs.
4	Launch	For 'turn on' or 'resume' power action if the VM is required to satisfy the requested desktop or app resource and is not in a suitable power state.
5	Admin	An administrator can directly request any power action (turn on, turn off, suspend, resume, restart, reboot etc.)
6	Untaint	For shutdown power action to 'clean' a VM OS disk back to the state of the golden master image that the VM was originally provisioned with.
7	Policy	Rules can be created to cause VMs to be shutdown or suspended some number of minutes after trigger events such as user disconnect or user logoff. This source is also used for some other cases, such as the shutdown issued to a VM if it doesn't register in time after a turn on.

Error code	Enum value	Description
8	IdlePool	For shutdown or turn on power actions generated to cause the appropriate number of VMs to be made running/ready at any one time, driven by the pool size and buffer size (and peak/off-peak times and timezone) configuration.

PowerActionTypeCode

Error code	Enum value	Description
0	Unknown	Unknown action type.
1	PowerOn	Turn a VM on from an 'off' state.
2	PowerOff	Abruptly turn off a VM without allowing the guest OS to shut down.
3	Shutdown	Gracefully shut down a VM, allowing the guest OS to perform shut down actions as needed.
4	Reset	Abruptly reset (turn off and then turn on) a VM without allowing the guest OS to shut down.
5	Restart	Gracefully restart (shut down and then turn on) a VM, allowing the guest OS to perform shut down actions as needed.
6	Suspend	Request a hypervisor suspend VM activity. This may or may not involve guest OS notification depending on hypervisor type.

Error code	Enum value	Description
7	Resume	Request a hypervisor resume a previously suspend VM activity. This may or may not involve guest OS notification depending on hypervisor type.

ProvisioningType

Error code	Enum value	Description
0	Unknown	Unknown.
1	MCS	Machine provisioned by Machine Creation Services (machine must be a VM).
2	PVS	Machine provisioned by Provisioning Services (may be physical, blade, VM).
3	Manual	No automated provisioning.

PowerStateCode

Error code	Enum value	Description
0	Unknown	Unknown power state.
1	Unavailable	Power state is not available.
2	Off	Machine is powered off.
3	On	Machine is powered on.
4	Suspended	Machine is suspended.
5	TurningOn	Machine is in the process of turning on.
6	TurningOff	Machine is in the process of turning off.

Error code	Enum value	Description
7	Suspending	Machine is in the process of suspending.
8	Resuming	Machine is in the process of resuming from suspension.
9	Unmanaged	Machine is not under management control.
10	NotSupported	Power state is not supported.
11	VirtualMachineNotFound	Virtual machine is not found.

PersistentUserChangesType

Error code	Enum value	Description
0	Unknown	Unknown.
1	Discard	User changes are discarded.
2	OnLocal	User changes are persisted locally.
3	OnPvd	User changes are persisted on the Pvd.

RegistrationState

Error code	Enum value	Description
0	Unknown	Unknown.
1	Registered	Machine is currently registered.
2	Unregistered	Machine has been unregistered.

SessionType

Error code	Enum value	Description
0	Desktop	Desktop session.
1	Application	Application session.

SessionSupportCode

Error code	Enum value	Description
0	Unknown	Unknown session support type.
1	SingleSession	Supports single session only (VDI).
2	MultiSession	Supports multiple concurrent sessions (RDS).

CostType

Type	Enum value	Description
0	Unknown	Unknown.
1	AssumedCost	Assumed Cost.
2	ConfiguredCostFromStudio	The cost configured in Studio.
3	AzureRetailAPI	Azure Retail API cost.
4	CostFromHostService	Cost associated with host service.
5	ActualCost	Actual Cost.

CurrencyType

Type	Enum value	Description
0	Unknown	Unknown.
1	USD	Currency type in US dollar.

Type	Enum value	Description
2	EUR	Euro currency in Europe.
3	CAD	Canadian dollar currency type.
4	CNY	Chinese yuan currency type.
5	BRL	Brazilian real currency type.
6	NZD	New Zealand dollar currency.
7	GBP	British pound currency type.
8	JPY	Japanese yen currency type.
9	KRW	South Korean won currency.

CostSavingsSummaryGranularity

Granularity	Enum value	Description
0	Hour	Granularity used for hour level data.
1	Day	Granularity used for day level data.
2	Month	Granularity used for month level data.
3	DayLevelInMinutes	Granularity used for day level data in terms of minutes.

HypervisorType

Granularity	Enum value	Description
0	None	Used for machine data for which there is no Hypervisor attached.
1	Unknown	Hypervisor type is unknown.
2	Azure	Hypervisor type from Microsoft.
3	AWS	Amazon Web Services hypervisor type.

Granularity	Enum value	Description
4	GCP	Hypervisor type from Google cloud.
5	XenServer	Hypervisor type from Citrix.
6	HyperV	Microsoft HyperV hypervisor type.
7	AHV	Hypervisor type from Nutanix.
8	VSphere	Hypervisor type from VMWare.

WorkspaceType

Error code	Enum value	Description
0	Unknown	Workspace type is Unknown.
1	Windows	Workspace type is Windows.
2	Mac	Workspace type is Mac.
3	Linux	Workspace type is Linux.
4	HTML5	Workspace type is HTML5.
5	Chrome	Workspace type is Chrome.

Best Practices for performance optimization

July 29, 2024

Optimizing API calls has been simplified through the effective implementation of parameters within the URL. Here are some useful samples.

Specifying a Custom Time Range

Using the powerful `$filter` function, you can tailor your queries to retrieve data within preset time frames. This results in enhanced efficiency due to reduced network traffic and faster processing.

Be mindful that time-related filtering is typically facilitated using the `ModifiedDate` field, which is prevalent across most entities. For those entities without such a field, kindly consult our comprehensive [API Exploration](#) guide for alternative filter options.

Request

```
1 GET /monitorodata/Connections?$filter=(ModifiedDate ge 2024-06-04T00:00:00Z and ModifiedDate le 2024-06-05T00:00:00Z) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Top N records

In scenarios where only a limited number of sample records are required, it's unnecessary to retrieve the entire dataset from the current entity.

Request

```
1 GET /monitorodata/Machines?$orderby=Name&$top=10 HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Best Practices for performance optimization

July 29, 2024

Optimizing API calls has been simplified through the effective implementation of parameters within the URL. Here are some useful samples.

Specifying a Custom Time Range

Using the powerful `$filter` function, you can tailor your queries to retrieve data within preset time frames. This results in enhanced efficiency due to reduced network traffic and faster processing.

Be mindful that time-related filtering is typically facilitated using the `ModifiedDate` field, which is prevalent across most entities. For those entities without such a field, kindly consult our comprehensive [API Exploration](#) guide for alternative filter options.

Request

```
1 GET /monitorodata/Connections?$filter=(ModifiedDate ge 2024-06-04T00:00:00Z and ModifiedDate le 2024-06-05T00:00:00Z) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Top N records

In scenarios where only a limited number of sample records are required, it's unnecessary to retrieve the entire dataset from the current entity.

Request

```
1 GET /monitorodata/Machines?$orderby=Name&$top=10 HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

How to invoke API

December 20, 2024

Use the [API Exploration](#) for a quick demo.

All the query parameters are available for other access methods mentioned in [Data Access Methods](#)

Example 1: step by step to invoke the API

Use the [Get Sessions](#) as an example.

Sessions GET

Get all Sessions

Resource URL

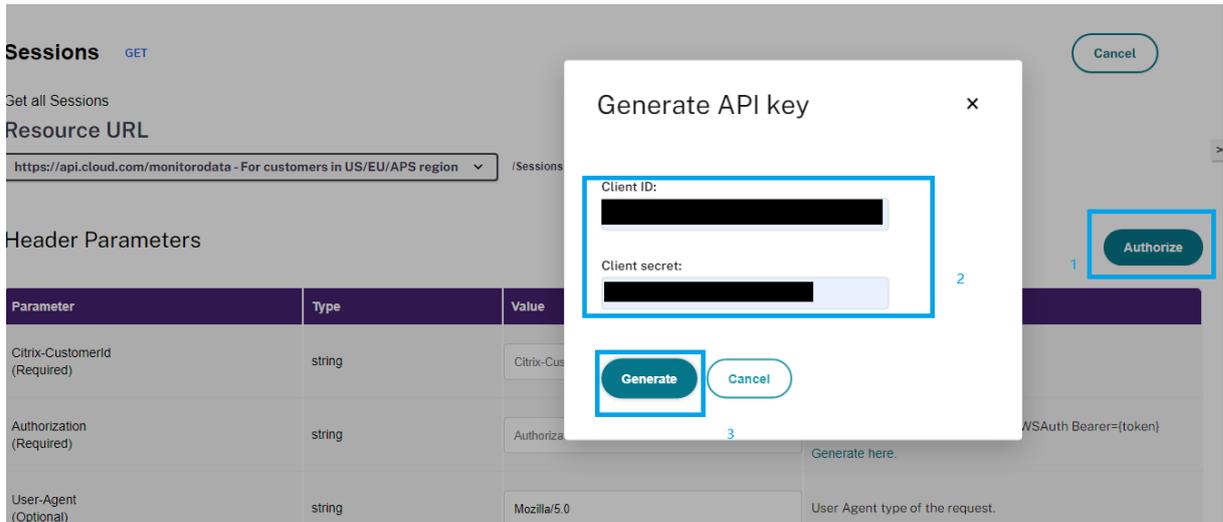
/Sessions

Header Parameters

Parameter	Type	Value	Description
Citrix-Customerid	string	Required	Citrix Cloud Customer ID.



Step 1. Choose the URL based on your region and click [Invoke API](#).



Step 2. Click [Authorize](#) and fill in the [Client ID](#) and [Client secret](#), then click the [Generate](#).

Refer to the [Get started with Citrix Cloud APIs](#) section to create the [Client ID](#) and [Client secret](#).

Authorization



Cloud auth token

Name: Authorization

In: header

Key value:

CwsAuth bearer=eyJhbGciOiJSUzI1NiIsI...

Note: The authorization key here is automatically appended to the API authentication request.

[Authorize](#)

[Back](#)

Step 3. Click [Authorize](#). The authorization key is automatically appended to the API authentication request.

Header Parameters

[Edit Authorization](#)

Parameter	Type	Value	Description
Citrix-CustomerId (Required)	string	<input type="text" value=""/>	Citrix Cloud Customer ID.
Authorization (Required)	string	<input type="text" value="CwsAuth bearer=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpX"/>	Citrix Cloud authorization header: CWSAuth Bearer={token} Generate here.
User-Agent (Optional)	string	<input type="text" value="Mozilla/5.0"/>	User Agent type of the request.
Accept (Optional)	string	<input type="text" value="application/json"/>	Must accept application/json.
Accept-Encoding (Optional)	string	<input type="text" value="gzip, deflate, br"/>	Must accept Encoding format.

Step 4. Fill in the header parameters.

- **Citrix-CustomerId:** Your customerID
- **Authorization:** Automatically appended in Step3, no need to change.
- **User-Agent:** Automatically filled, no need to change.
- **Accept:** Automatically filled, no need to change.
- **Accept-Encoding:** Automatically filled, no need to change. **Encoding can reduce bandwidth, saving on costs.**

Parameter	Type	Value	Description
\$filter (Optional)	string	<input type="text" value="\$StartDate gt 2024-01-01T00:00:00Z and StartDate lt 2"/>	Filter the results using OData syntax. Please use the indexed fields for the filter to improve the performance. Indexed fields: CurrentConnectionId, FailureId, UserId, PublishedDesktopId, MachineId, LifecycleState, SessionType, EndDate, StartDate, ConnectionState.
\$orderby (Optional)	string	<input type="text" value="\$orderby - Order the results using OData syntax."/>	Order the results using OData syntax.
\$top (Optional)	integer	<input type="text" value="1"/>	Limit the number of results returned.
\$skip (Optional)	integer	<input type="text" value="\$skip - Skip the number of results."/>	Skip the number of results.
\$count (Optional)	boolean	<input type="checkbox" value="false"/>	Include the count of results.
\$select (Optional)	string	<input type="text" value="\$select - Select the fields to return."/>	Select the fields to return.
\$expand (Optional)	string	<input type="text" value="\$expand - Expand the fields to return. The example v"/>	Expand the fields to return. The example value and schema in the response show the result after using the expand operation. Without the expand operation, only return the current entity.

Step 5. Fill in the query parameters.

- **\$filter:** Filter the results using OData syntax. **Please use the indexed fields for better performance.**
- **\$orderby:** Order the results using OData syntax.
- **\$top:** Limit the number of results returned. Here is 1 for sample, please adjust or remove it as

needed.

- **\$skip**: Skip the number of results.
- **\$count**: Include the count of results.
- **\$select**: Select the fields to return. **Use this operation to return the required fields, reducing bandwidth and saving on costs.**
- **\$expand**: Expand the fields to return. The example value and schema in the response show the result after using the expand operation. Without the expand operation, only the current entity will be displayed, and the related entities will not be shown.



Responses

curl -X "GET" \
 "https://api.cloudburrito.com/monitorodata/Sessions?filter=StartDate%202024-01-01T00:34:00.3400Z&bandwidthStartdate%201%202024-12-02T00:34:00.3400Z&top=1&count=false" \
 -H "accept: */*" \
 -H "Citrix-CustomerId: " \
 -H "User-Agent: Mozilla/5.0" \
 -H "Accept-Encoding: gzip, deflate, br" \
 -H "Authorization: OAuth bearer: "

Request URL

https://api.cloudburrito.com/monitorodata/Sessions?filter=StartDate%202024-01-01T00:34:00.3400Z&bandwidthStartdate%201%202024-12-02T00:34:00.3400Z&top=1&count=false

Server response

Code 200 Details

```

{
  "@odata.context": "https://api.cloudburrito.com/monitorodata/$metadata#Sessions",
  "value": [
    {
      "SessionKey": "Adc26a32-f97d-4f6e-82d9-85184a68419f",
      "StartDate": "2024-09-13T05:12:52.073Z",
      "LogonDuration": 29423,
      "EndDate": "2024-09-13T16:34:03.893Z",
      "ExitCode": 1,
      "FailureId": null,
      "FailureCode": null,
      "ConnectionState": 3,
      "SessionIdleTime": "2024-09-13T06:43:47Z",
      "ConnectionStateChangeDate": "2024-09-13T16:34:03.893Z",
      "ErrorCode": null,
      "CurrentConnectionId": 8,
      "User": null
    }
  ]
}
    
```

Response headers

```

alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
content-encoding: gzip
content-length: 408
content-type: application/json; odata.metadata=minimal
date: Tue, 20 Dec 2024 09:10:42 GMT
function-exception-id: 0d6bee95xw
odata-version: 4.0
server: Google frontent
strict-transport-security: max-age=31536000
x-cloud-trace-context: b5961400b02e95d5c932295fadacef0f
x-powered-by: Express
x-ratelimit-limit: 30000000
x-ratelimit-remaining: 30000000
x-ratelimit-reset: 128
    
```

Step 6. Click the **execute** then check the reponse.

All the query parameters will be shown in the **Request URL**

Example 2: the \$count operation

With **\$count=true** in the query parameter.

\$count (Optional)	boolean	true	Include the count of results.
--------------------	---------	------	-------------------------------

The response will contains the count filed **@odata.count**.

Request URL

```
https://api.cloudburrito.com/monitorodata/Sessions?%24filter=StartDate%20gt%202024-01-01T00%3A00%3A00Z%20and%20StartDate%20lt%202024-12-02T00%3A00%3A00Z%24top=18%24count=true
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "@odata.context": "https://api.cloudburrito.com/monitorodata/\$metadata#Sessions", "@odata.count": 4, "value": [{ "SessionKey": "4dc26a32-f97d-4f6e-82d9-85184a68419f", "StartDate": "2024-09-13T05:12:52.073Z", "LogOnDuration": 29427, "EndDate": "2024-09-13T16:34:03.893Z", "ExitCode": 1, "FailureId": null, }] }</pre>

Example 3: the \$select operation

With \$select in the query parameter.

\$select (Optional)	string	SessionKey,StartDate,MachineId	Select the fields to return.
------------------------	--------	--------------------------------	------------------------------

The response will only return the selected files.

```
https://api.cloudburrito.com/monitorodata/Sessions?%24filter=StartDate%20gt%202024-01-01T00%3A00%3A00Z%20and%20StartDate%20lt%202024-12-02T00%3A00%3A00Z%24top=18%24count=true&%24select=SessionKey%2CStartDate%2CMachineId
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "@odata.context": "https://api.cloudburrito.com/monitorodata/\$metadata#Sessions(SessionKey,StartDate,MachineId)", "@odata.count": 4, "value": [{ "SessionKey": "4dc26a32-f97d-4f6e-82d9-85184a68419f", "StartDate": "2024-09-13T05:12:52.073Z", "MachineId": "faaf6196-0fc2-41d3-af93-9b9f60047d47" }] }</pre>

Example 4: the \$expand operation

With \$expand in the query parameter.

\$expand (Optional)	string	Machine	Expand the fields to return. The example value and schema in the response show the result after using the expand operation. Without the expand operation, only return the current entity.
------------------------	--------	---------	---

The response only return the expand files.

https://api.cloudburrito.com/monitorodata/Sessions?%24filter=StartDate%20gt%202024-01-01%20and%20StartDate%20lt%202024-12-02%20and%20count=true%24select=SessionKey%2CStartDate%2CMachineId%2Cexpand=Machine

ver response

Code	Details
200	<p>Response body</p> <pre>{ "@odata.context": "https://api.cloudburrito.com/monitorodata/\$metadata#Sessions(SessionKey,StartDate,MachineId,Machine())", "@odata.count": 4, "value": [{ "SessionKey": "4dc26a32-f97d-4f6e-82d9-85184a68419f", "StartDate": "2024-09-13T05:12:52.073Z", "MachineId": "faaf6196-0fc2-41d3-af93-9b9f60047d47", "Machine": { "Id": "faaf6196-0fc2-41d3-af93-9b9f60047d47", "Sid": "S-1-5-21-792844889-570118499-417503878-48601", "Name": "INTESCALE\\formalstg01", "DnsName": "formalstg01.intescale.local", "LifecycleState": 1, "IPAddress": null, "HostedMachineId": "intescale-rg/formalstg01", "HostedMachineName": "" } }] }</pre>

Machine data request examples

June 2, 2025

The Machines entity represents physical or virtual machines in the customer's cloud or on-prem Site. Use Citrix Monitor Service API to get machine data.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/Machines` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

Active machines with count

- `LifecycleState` value `0` represents Active machines.

- `$count=true` parameter appends the record count in the response.

Request

```
1 GET /monitorodata/Machines?$filter=LifecycleState eq 0&$count=true HTTP
  /1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Machines",
11   "@odata.count": 517,
12   "value": [
13     {
14
15       "Id": "17f8b8f5-0f27-4852-a399-00194801a283",
16       "Sid": "S-1-5-21-1924580635-3864722089-3907235813-107439",
17       "Name": "ENG\\ra-s19-cvad036",
18       "DnsName": "ra-s19-cvad036.eng.citrite.net",
19       "LifecycleState": 0,
20       "IPAddress": null,
21       "HostedMachineId": "ccf63697-723c-0845-17fe-ebe1e2e75d31",
22       "HostingServerName": "ra-xen-07",
23       "HostedMachineName": "ra-s19-cvad036",
24       "IsAssigned": false,
25       "IsInMaintenanceMode": false,
26       "IsPendingUpdate": false,
27       "AgentVersion": "2212.0.0.77",
28       "AssociatedUserFullNames": "",
29       "AssociatedUserNames": "",
30       "AssociatedUserUPNs": "",
31       "CurrentRegistrationState": 2,
32       "RegistrationStateChangeDate": "2024-08-05T22:03:08.31Z",
33       "LastDeregisteredCode": 0,
34       "LastDeregisteredDate": "2024-08-05T22:02:33.64Z",
35       "CurrentPowerState": 2,
36       "CurrentSessionCount": 0,
37       "ControllerDnsName": "A21828-46-2.prodcp5.local",
```

```

38     "PoweredOnDate": "2024-08-05T10:01:50.4Z",
39     "PowerStateChangeDate": "2024-08-05T22:03:08.31Z",
40     "FunctionalLevel": 10,
41     "FailureDate": null,
42     "WindowsConnectionSetting": 0,
43     "IsPreparing": false,
44     "FaultState": 1,
45     "OSType": "Windows 2019",
46     "CurrentLoadIndexId": 17863392,
47     "CatalogId": "11c5298e-4516-4c1b-b83a-efb219009b71",
48     "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",
49     "HypervisorId": "56ac3d9f-2691-4d2c-aacb-95726611fe79",
50     "LastPowerActionType": 3,
51     "LastPowerActionReason": 8,
52     "LastPowerActionFailureReason": 400,
53     "LastPowerActionCompletedDate": "2024-08-05T22:02:50.887Z",
54     "Tags": [],
55     "Hash": "",
56     "MachineRole": 0,
57     "CreatedDate": "2020-06-10T14:31:36.183Z",
58     "ModifiedDate": "2024-08-05T22:47:00.103Z"
59   }
60 ]
61 }
62 }

```

Machines in a failure state

- `FailureDate` is `null` represents machines not in a failure state.
- `FailureDate` ne `null` represents machines in a failure state.
- `FailureDate` ge 2024-07-30T00:00:00Z helps filter machines in a failure state after 2024-07-30T00:00:00Z.

Request

```

1 GET /monitorodata/Machines?null=null&$filter=FailureDate ne null and
   FailureDate ge 2024-07-30T00:00:00Z HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal

```

```
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     Machines",
12     "value": [
13         {
14             "Id": "70fe7ae3-e933-4f99-aabf-07153c261ee7",
15             "Sid": "S-1-5-21-1924580635-3864722089-3907235813-416405",
16             "Name": "ENG\\ra-w10-cvad008",
17             "DnsName": "ra-w10-cvad008.eng.citrite.net",
18             "LifecycleState": 0,
19             "IPAddress": null,
20             "HostedMachineId": "17934eda-3307-16a1-a7ee-0594d5f65a89",
21             "HostingServerName": "ra-xen-07",
22             "HostedMachineName": "ra-w10-cvad008",
23             "IsAssigned": false,
24             "IsInMaintenanceMode": false,
25             "IsPendingUpdate": false,
26             "AgentVersion": "2203.0.0.33220",
27             "AssociatedUserFullNames": "",
28             "AssociatedUserNames": "",
29             "AssociatedUserUPNs": "",
30             "CurrentRegistrationState": 2,
31             "RegistrationStateChangeDate": "2024-07-30T23:49:47.18Z",
32             "LastDeregisteredCode": 207,
33             "LastDeregisteredDate": "2024-07-30T23:49:47.18Z",
34             "CurrentPowerState": 3,
35             "CurrentSessionCount": 0,
36             "ControllerDnsName": "A21828-46-1.prodcp5.local",
37             "PoweredOnDate": "2024-07-30T23:35:33.757Z",
38             "PowerStateChangeDate": "2024-07-31T23:45:16.86Z",
39             "FunctionalLevel": 9,
40             "FailureDate": "2024-07-30T23:49:47.18Z",
41             "WindowsConnectionSetting": 0,
42             "IsPreparing": false,
43             "FaultState": 4,
44             "OSType": "Windows 10",
45             "CurrentLoadIndexId": 17863154,
46             "CatalogId": "8979ff21-701b-4cf6-bb3f-18a77cf46427",
47             "DesktopGroupId": "87f4c976-b8f8-4b0e-82c3-dcd7f94c3661",
48             "HypervisorId": "56ac3d9f-2691-4d2c-aacb-95726611fe79",
49             "LastPowerActionType": 3,
50             "LastPowerActionReason": 8,
51             "LastPowerActionFailureReason": 401,
52             "LastPowerActionCompletedDate": "2024-07-31T23:45:16.827Z",
53             "Tags": [],
54             "Hash": "",
55             "MachineRole": 0,
```

```
56         "CreatedDate": "2022-09-28T18:16:45.633Z",
57         "ModifiedDate": "2024-07-31T23:47:00.173Z"
58     }
59
60 ]
61 }
```

Get Machine failure details

- The `MachineFailures` entity provides details about Machine failures. Use the `$expand` operation to enrich the response records with these details.

Request

```
1 GET /monitorodata/Machines?$filter=FailureDate ne null and FailureDate
   ge 2024-07-30T00:00:00Z&$select=Id&$expand=MachineFailures($filter=
   FailureStartDate ge 2024-07-30T00:00:00Z) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Machines",
11     "value": [
12         {
13
14             "Id": "70fe7ae3-e933-4f99-aabf-07153c261ee7",
15             "MachineFailures": [
16                 {
17
18                     "Id": 1589020,
19                     "MachineId": "70fe7ae3-e933-4f99-aabf-07153c261ee7"
20                     ,
21                     "FailureStartDate": "2024-07-30T11:13:43.04Z",
22                     "FailureEndDate": "2024-07-30T23:34:55.38Z",
23                     "FaultState": 4,
```

```

23         "LastDeregisteredCode": 207,
24         "CreatedDate": "2024-07-30T11:13:54.95Z",
25         "ModifiedDate": "2024-07-30T23:34:59.7Z"
26     }
27 ,
28     {
29
30         "Id": 1589059,
31         "MachineId": "70fe7ae3-e933-4f99-aabf-07153c261ee7"
32     ,
33         "FailureStartDate": "2024-07-30T23:49:47.18Z",
34         "FailureEndDate": null,
35         "FaultState": 4,
36         "LastDeregisteredCode": 207,
37         "CreatedDate": "2024-07-30T23:49:49.517Z",
38         "ModifiedDate": "2024-07-30T23:49:49.517Z"
39     }
40 ]
41 }
42
43 ]
44 }

```

Find failed machine power actions

- The `LastPowerActionReason` and `LastPowerActionFailureReason` fields can be used to filter for the failed machine power actions. Check page [Descriptions for enums and error codes](#) for details.

Request

```

1 GET /monitorodata/Machines?$filter=LifecycleState eq 0 and DesktopGroup
  /SessionSupport eq 1 and (cast(LastPowerActionReason,'Edm.Int32') eq
  6 or cast(LastPowerActionReason,'Edm.Int32') eq 7 or cast(
  LastPowerActionReason,'Edm.Int32') eq 8) and (cast(
  LastPowerActionFailureReason,'Edm.Int32') eq 300 or cast(
  LastPowerActionFailureReason,'Edm.Int32') eq 402 or cast(
  LastPowerActionFailureReason,'Edm.Int32') eq 401) and MachineRole eq
  0&$orderby=Name&$skip=0&$top=50&$expand=DesktopGroup($select=Name)&
  $select=Sid,Id,Name,IsAssigned,IPAddress,FaultState,
  LastDeregisteredCode,FailureDate,CurrentPowerState,
  CurrentSessionCount,IsInMaintenanceMode,LastPowerActionFailureReason
  ,LastPowerActionCompletedDate HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     Machines",
12     "value": [
13         {
14             "Id": "a9db1234-16f6-45c2-bb1b-a6f978fe9810",
15             "Sid": "S-1-5-21-2061182105-2775619996-3779015820-12345",
16             "Name": "EWS\\LR0003",
17             "IPAddress": null,
18             "IsAssigned": true,
19             "IsInMaintenanceMode": false,
20             "LastDeregisteredCode": 2,
21             "CurrentPowerState": 2,
22             "CurrentSessionCount": 0,
23             "FailureDate": null,
24             "FaultState": 1,
25             "LastPowerActionFailureReason": 401,
26             "LastPowerActionCompletedDate": "2025-05-13T04:59:50.18Z",
27             "DesktopGroup": {
28
29                 "Name": "LRO-2-P"
30             }
31         }
32     ]
33 }
34 ]
35 }
```

Alive sessions on each machine

- `Sessions` in the `$expand` operation enriches the response records with the sessions running on a machine.
- `EndDate eq null` represents the sessions that are still alive.

```
1 GET /monitorodata/Machines?$select=Name,IsInMaintenanceMode,
   CurrentRegistrationState,CurrentSessionCount,IsPreparing,FaultState&
   $expand=DesktopGroup($select=Id,Name),Sessions($select=SessionKey,
   ConnectionState;$filter=EndDate+eq+null)&$filter=Name+ne+null+and+
   DesktopGroup+ne+null HTTP/1.1
```

```
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 122546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     Machines",
12     "value": [
13         {
14             "Name": "ENG\\ra-s19-cvad079",
15             "IsInMaintenanceMode": false,
16             "CurrentRegistrationState": 1,
17             "CurrentSessionCount": 1,
18             "IsPreparing": false,
19             "FaultState": 1,
20             "DesktopGroup": {
21
22                 "Id": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",
23                 "Name": "CVD\\BUR CVAD Development"
24             }
25         },
26         {
27             "Sessions": [
28                 {
29                     "SessionKey": "d0d8b5ab-9450-4beb-9f3e-9f7c1577d9bd",
30                     "ConnectionState": 2
31                 }
32             ]
33         }
34     ]
35 }
36 ]
37 }
```

Active VDA machines associated information

- `LifecycleState eq 0` represents active machines.
- `MachineRole eq 0` represents a VDA machine.
- Use `$expand` to get associated information like catalog, hypervisor and load.

Request

```
1 GET /monitorodata/Machines?$filter=LifecycleState eq 0 and Name ne null
   and DnsName ne '' and CatalogId ne null and MachineRole eq 0&
   $expand=CurrentLoadIndex,Catalog,Hypervisor HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 122546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Machines",
11     "value": [
12         {
13
14             "Id": "17f8b8f5-0f27-4852-a399-00194801a283",
15             "Sid": "S-1-5-21-1924580635-3864722089-3907235813-107439",
16             "Name": "ENG\\ra-s19-cvad036",
17             "DnsName": "ra-s19-cvad036.eng.citrite.net",
18             "LifecycleState": 0,
19             "IPAddress": null,
20             "HostedMachineId": "ccf63697-723c-0845-17fe-ebe1e2e75d31",
21             "HostingServerName": "ra-xen-07",
22             "HostedMachineName": "ra-s19-cvad036",
23             "IsAssigned": false,
24             "IsInMaintenanceMode": false,
25             "IsPendingUpdate": false,
26             "AgentVersion": "2212.0.0.77",
27             "AssociatedUserFullNames": "",
28             "AssociatedUserNames": "",
29             "AssociatedUserUPNs": "",
30             "CurrentRegistrationState": 2,
31             "RegistrationStateChangeDate": "2024-08-05T22:03:08.31Z",
32             "LastDeregisteredCode": 0,
33             "LastDeregisteredDate": "2024-08-05T22:02:33.64Z",
34             "CurrentPowerState": 2,
35             "CurrentSessionCount": 0,
36             "ControllerDnsName": "A21828-46-2.prodcp5.local",
37             "PoweredOnDate": "2024-08-05T10:01:50.4Z",
38             "PowerStateChangeDate": "2024-08-05T22:03:08.31Z",
39             "FunctionalLevel": 10,
```

```
40     "FailureDate": null,
41     "WindowsConnectionSetting": 0,
42     "IsPreparing": false,
43     "FaultState": 1,
44     "OSType": "Windows 2019",
45     "CurrentLoadIndexId": 17863392,
46     "CatalogId": "11c5298e-4516-4c1b-b83a-efb219009b71",
47     "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",
48     "HypervisorId": "56ac3d9f-2691-4d2c-aacb-95726611fe79",
49     "LastPowerActionType": 3,
50     "LastPowerActionReason": 8,
51     "LastPowerActionFailureReason": 400,
52     "LastPowerActionCompletedDate": "2024-08-05T22:02:50.887Z",
53     "Tags": [],
54     "Hash": "",
55     "MachineRole": 0,
56     "CreatedDate": "2020-06-10T14:31:36.183Z",
57     "ModifiedDate": "2024-08-05T22:47:00.103Z",
58     "CurrentLoadIndex": {
59
60         "Id": 17863392,
61         "EffectiveLoadIndex": 0,
62         "Cpu": 0,
63         "Memory": 0,
64         "Disk": 0,
65         "Network": 0,
66         "SessionCount": 0,
67         "MachineId": "17f8b8f5-0f27-4852-a399-00194801a283",
68         "CreatedDate": "2024-08-05T22:03:12.277Z",
69         "ModifiedDate": "2024-08-06T06:51:36.9277153Z"
70     },
71     ,
72     "Catalog": {
73
74         "Id": "11c5298e-4516-4c1b-b83a-efb219009b71",
75         "Name": "CVD\\BUR CVAD Development",
76         "LifecycleState": 0,
77         "ProvisioningType": 2,
78         "PersistentUserChanges": 1,
79         "IsMachinePhysical": false,
80         "AllocationType": 2,
81         "SessionSupport": 1,
82         "ProvisioningSchemeId": null,
83         "CreatedDate": "2020-05-19T00:02:03.903Z",
84         "ModifiedDate": "2022-12-10T13:51:24.173Z"
85     },
86     ,
87     "Hypervisor": {
88
89         "Id": "56ac3d9f-2691-4d2c-aacb-95726611fe79",
90         "Name": "BUR_Prod0",
91         "Type": "XenServer",
92         "LifecycleState": 0,
```

```

93         "CreatedDate": "2020-05-15T12:32:30.587Z",
94         "ModifiedDate": "2024-03-02T00:07:05.303Z"
95     }
96
97 }
98
99 ]
100 }

```

Get VDA machine availability over time

- `MachineSummaries` entity provides aggregate machine metrics at hourly or daily granularity.
- The `$expand` operations enriches the records with Desktop Group information.

Request

```

1 GET /monitorodata/MachineSummaries?$filter=Granularity eq 60 and
    SummaryDate ge 2025-05-01&$select=SummaryDate,PoweredOnMachinesCount
    ,RegisteredMachinesCount,MachinesInMaintenanceModeCount,
    MachinesCount&$expand=DesktopGroup($select=Id,Name)&$orderby=
    DesktopGroup/Name HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 2546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Machines",
11     "value": [
12         {
13
14             "SummaryDate": "2025-05-01T00:00:00Z",
15             "PoweredOnMachinesCount": 7,
16             "RegisteredMachinesCount": 5,
17             "MachinesInMaintenanceModeCount": 0,
18             "MachinesCount": 8,

```

```

19         "DesktopGroup":{
20             "Id":"002683cd-b3a5-4fc8-913c-9f0d12e4d7cb",
21             "Name":"LAB3-W10-P"
22         }
23     },
24     {
25         "SummaryDate":"2025-05-01T01:00:00Z",
26         "PoweredOnMachinesCount":0,
27         "RegisteredMachinesCount":0,
28         "MachinesInMaintenanceModeCount":1,
29         "MachinesCount":2,
30         "DesktopGroup":{
31             "Id":"002683cd-b3a5-4fc8-913c-9f0d12e4d7cb",
32             "Name":"LAB3-W10-P"
33         }
34     }
35 ]
36 }

```

List of Hotfixes applied to each machine

As part of this example, we query the MachineHotfixLogs entity, which includes records with each of the hotfixes applied to the VDA machines. We enrich the result records with Machine and Desktop-Group information.

- `MachineHotfixLogs` provides a list of the hotfixes.
- Use `$expand` to get associated Machine, DesktopGroup and Hotfix information.

Request

```

1 GET /monitorodata/MachineHotfixLogs?$filter=CreatedDate ge 2025-02-01&
   $expand=Machine($select=Name;$expand=DesktopGroup($select=Name)),
   Hotfix&$orderby=Machine/DesktopGroup/Name,Machine/Name HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 2546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     Machines(CurrentLoadIndex(),Catalog(),Hypervisor())",
12     "value": [
13         {
14             "Id":10073,
15             "MachineId":"6829529e-4126-417c-90a7-a372dc7bcbc7",
16             "HotfixId":"33fd700d-d35f-4b6c-81e0-30cd793547f7",
17             "ChangeType":1,
18             "CurrentState":true,
19             "CreatedDate":"2024-11-27T13:24:38.05Z",
20             "ModifiedDate":"2024-11-27T13:24:38.05Z",
21             "Machine":{
22
23                 "Name":"LAB\\TEST-MVDA-01",
24                 "DesktopGroup":{
25
26                     "Name":"DG-W10-PMD"
27                 }
28             }
29         },
30         "Hotfix":{
31
32             "Id":"33fd700d-d35f-4b6c-81e0-30cd793547f7",
33             "Name":"IdentityAssertionWX64_7_33_5000",
34             "Article":"http://support.citrix.com/article/CTX579943"
35         },
36         "ArticleName":"CTX579943",
37         "FileName":"Citrix.Authentication.IdentityAssertion_x64",
38         "FileFormat":"MSI",
39         "Version":"10.9.5000.8",
40         "ComponentName":"Citrix Identity Assertion VDA Plugin",
41         "ComponentVersion":"10.9.5000.8",
42         "CreatedDate":"2024-06-07T19:42:49.38Z",
43         "ModifiedDate":"2024-06-07T19:42:49.38Z"
44     }
45 }
46 ]
47 ]
48 ]
49 }
```

Get a list of all the VDA versions

Query the Machines entity, aggregate over the distinct versions and calculate the number of machines running each of the VDA versions per delivery group.

Request

```
1 GET /monitorodata/Machines?$apply=filter(AgentVersion ne null and
   DesktopGroup/Name ne null)/expand(DesktopGroup,filter(Name ne null))
   /groupby((DesktopGroup/Name,AgentVersion),aggregate(Id with
   countdistinct as MachineCount)) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 2546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Machines(CurrentLoadIndex(),Catalog(),Hypervisor())",
11   "value": [
12     {
13
14       "@odata.id":null,
15       "AgentVersion":"2407.0.10.101",
16       "MachineCount":46,
17       "DesktopGroup":{
18
19         "@odata.id":null,
20         "Name":"DG-W10-LRO-P"
21       }
22     }
23   ],
24   {
25     {
26
27       "@odata.id":null,
28       "AgentVersion":"2402.0.100.609",
29       "MachineCount":11,
30       "DesktopGroup":{
31
```

```
32         "@odata.id":null,  
33         "Name":"DG-W10-PMD"  
34     }  
35  
36     }  
37  
38 ]  
39 }
```

Machine data request examples

June 2, 2025

The Machines entity represents physical or virtual machines in the customer's cloud or on-prem Site. Use Citrix Monitor Service API to get machine data.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/Machines` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

Active machines with count

- **LifecycleState** value 0 represents Active machines.
- **\$count=true** parameter appends the record count in the response.

Request

```
1 GET /monitorodata/Machines?$filter=LifecycleState eq 0&$count=true HTTP  
  /1.1  
2 Host: api.cloud.com
```

```
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11         Machines",
12     "@odata.count": 517,
13     "value": [
14         {
15             "Id": "17f8b8f5-0f27-4852-a399-00194801a283",
16             "Sid": "S-1-5-21-1924580635-3864722089-3907235813-107439",
17             "Name": "ENG\\ra-s19-cvad036",
18             "DnsName": "ra-s19-cvad036.eng.citrite.net",
19             "LifecycleState": 0,
20             "IPAddress": null,
21             "HostedMachineId": "ccf63697-723c-0845-17fe-ebe1e2e75d31",
22             "HostingServerName": "ra-xen-07",
23             "HostedMachineName": "ra-s19-cvad036",
24             "IsAssigned": false,
25             "IsInMaintenanceMode": false,
26             "IsPendingUpdate": false,
27             "AgentVersion": "2212.0.0.77",
28             "AssociatedUserFullNames": "",
29             "AssociatedUserNames": "",
30             "AssociatedUserUPNs": "",
31             "CurrentRegistrationState": 2,
32             "RegistrationStateChangeDate": "2024-08-05T22:03:08.31Z",
33             "LastDeregisteredCode": 0,
34             "LastDeregisteredDate": "2024-08-05T22:02:33.64Z",
35             "CurrentPowerState": 2,
36             "CurrentSessionCount": 0,
37             "ControllerDnsName": "A21828-46-2.prodcp5.local",
38             "PoweredOnDate": "2024-08-05T10:01:50.4Z",
39             "PowerStateChangeDate": "2024-08-05T22:03:08.31Z",
40             "FunctionalLevel": 10,
41             "FailureDate": null,
42             "WindowsConnectionSetting": 0,
43             "IsPreparing": false,
44             "FaultState": 1,
45             "OSType": "Windows 2019",
```

```
46     "CurrentLoadIndexId": 17863392,  
47     "CatalogId": "11c5298e-4516-4c1b-b83a-efb219009b71",  
48     "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",  
49     "HypervisorId": "56ac3d9f-2691-4d2c-aacb-95726611fe79",  
50     "LastPowerActionType": 3,  
51     "LastPowerActionReason": 8,  
52     "LastPowerActionFailureReason": 400,  
53     "LastPowerActionCompletedDate": "2024-08-05T22:02:50.887Z",  
54     "Tags": [],  
55     "Hash": "",  
56     "MachineRole": 0,  
57     "CreatedDate": "2020-06-10T14:31:36.183Z",  
58     "ModifiedDate": "2024-08-05T22:47:00.103Z"  
59   }  
60 ]  
61 }  
62 }
```

Machines in a failure state

- `FailureDate` is `null` represents machines not in a failure state.
- `FailureDate` ne `null` represents machines in a failure state.
- `FailureDate` ge 2024-07-30T00:00:00Z helps filter machines in a failure state after 2024-07-30T00:00:00Z.

Request

```
1 GET /monitorodata/Machines?null=null&$filter=FailureDate ne null and  
   FailureDate ge 2024-07-30T00:00:00Z HTTP/1.1  
2 Host: api.cloud.com  
3 Authorization: CWSAuth bearer=<Token>  
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK  
2 Date: Thu, 01 Aug 2024 06:23:00 GMT  
3 Content-Type: application/json; odata.metadata=minimal  
4 Content-Length: 5506  
5 Connection: keep-alive  
6 odata-version: 4.0  
7  
8 {  
9  
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#  
    Machines",
```

```
11     "value": [  
12         {  
13  
14             "Id": "70fe7ae3-e933-4f99-aabf-07153c261ee7",  
15             "Sid": "S-1-5-21-1924580635-3864722089-3907235813-416405",  
16             "Name": "ENG\\ra-w10-cvad008",  
17             "DnsName": "ra-w10-cvad008.eng.citrite.net",  
18             "LifecycleState": 0,  
19             "IPAddress": null,  
20             "HostedMachineId": "17934eda-3307-16a1-a7ee-0594d5f65a89",  
21             "HostingServerName": "ra-xen-07",  
22             "HostedMachineName": "ra-w10-cvad008",  
23             "IsAssigned": false,  
24             "IsInMaintenanceMode": false,  
25             "IsPendingUpdate": false,  
26             "AgentVersion": "2203.0.0.33220",  
27             "AssociatedUserFullNames": "",  
28             "AssociatedUserNames": "",  
29             "AssociatedUserUPNs": "",  
30             "CurrentRegistrationState": 2,  
31             "RegistrationStateChangeDate": "2024-07-30T23:49:47.18Z",  
32             "LastDeregisteredCode": 207,  
33             "LastDeregisteredDate": "2024-07-30T23:49:47.18Z",  
34             "CurrentPowerState": 3,  
35             "CurrentSessionCount": 0,  
36             "ControllerDnsName": "A21828-46-1.prodcp5.local",  
37             "PoweredOnDate": "2024-07-30T23:35:33.757Z",  
38             "PowerStateChangeDate": "2024-07-31T23:45:16.86Z",  
39             "FunctionalLevel": 9,  
40             "FailureDate": "2024-07-30T23:49:47.18Z",  
41             "WindowsConnectionSetting": 0,  
42             "IsPreparing": false,  
43             "FaultState": 4,  
44             "OSType": "Windows 10",  
45             "CurrentLoadIndexId": 17863154,  
46             "CatalogId": "8979ff21-701b-4cf6-bb3f-18a77cf46427",  
47             "DesktopGroupId": "87f4c976-b8f8-4b0e-82c3-dcd7f94c3661",  
48             "HypervisorId": "56ac3d9f-2691-4d2c-aacb-95726611fe79",  
49             "LastPowerActionType": 3,  
50             "LastPowerActionReason": 8,  
51             "LastPowerActionFailureReason": 401,  
52             "LastPowerActionCompletedDate": "2024-07-31T23:45:16.827Z",  
53             "Tags": [],  
54             "Hash": "",  
55             "MachineRole": 0,  
56             "CreatedDate": "2022-09-28T18:16:45.633Z",  
57             "ModifiedDate": "2024-07-31T23:47:00.173Z"  
58         }  
59     ]  
60 }  
61 }
```

Get Machine failure details

- The `MachineFailures` entity provides details about Machine failures. Use the `$expand` operation to enrich the response records with these details.

Request

```
1 GET /monitorodata/Machines?$filter=FailureDate ne null and FailureDate
  ge 2024-07-30T00:00:00Z&$select=Id&$expand=MachineFailures($filter=
  FailureStartDate ge 2024-07-30T00:00:00Z) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
  Machines",
11   "value": [
12     {
13
14       "Id": "70fe7ae3-e933-4f99-aabf-07153c261ee7",
15       "MachineFailures": [
16         {
17
18           "Id": 1589020,
19           "MachineId": "70fe7ae3-e933-4f99-aabf-07153c261ee7"
20           ,
21           "FailureStartDate": "2024-07-30T11:13:43.04Z",
22           "FailureEndDate": "2024-07-30T23:34:55.38Z",
23           "FaultState": 4,
24           "LastDeregisteredCode": 207,
25           "CreatedDate": "2024-07-30T11:13:54.95Z",
26           "ModifiedDate": "2024-07-30T23:34:59.7Z"
27         }
28       ]
29     }
30     ,
31     {
32
33       "Id": 1589059,
```

```

31         "MachineId": "70fe7ae3-e933-4f99-aabf-07153c261ee7"
32         ,
33         "FailureStartDate": "2024-07-30T23:49:47.18Z",
34         "FailureEndDate": null,
35         "FaultState": 4,
36         "LastDeregisteredCode": 207,
37         "CreatedDate": "2024-07-30T23:49:49.517Z",
38         "ModifiedDate": "2024-07-30T23:49:49.517Z"
39     }
40 ]
41 }
42
43 ]
44 }

```

Find failed machine power actions

- The `LastPowerActionReason` and `LastPowerActionFailureReason` fields can be used to filter for the failed machine power actions. Check page [Descriptions for enums and error codes](#) for details.

Request

```

1 GET /monitorodata/Machines?$filter=LifecycleState eq 0 and DesktopGroup
  /SessionSupport eq 1 and (cast(LastPowerActionReason,'Edm.Int32') eq
  6 or cast(LastPowerActionReason,'Edm.Int32') eq 7 or cast(
  LastPowerActionReason,'Edm.Int32') eq 8) and (cast(
  LastPowerActionFailureReason,'Edm.Int32') eq 300 or cast(
  LastPowerActionFailureReason,'Edm.Int32') eq 402 or cast(
  LastPowerActionFailureReason,'Edm.Int32') eq 401) and MachineRole eq
  0&$orderby=Name&$skip=0&$top=50&$expand=DesktopGroup($select=Name)&
  $select=Sid,Id,Name,IsAssigned,IPAddress,FaultState,
  LastDeregisteredCode,FailureDate,CurrentPowerState,
  CurrentSessionCount,IsInMaintenanceMode,LastPowerActionFailureReason
  ,LastPowerActionCompletedDate HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive

```

```
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
      Machines",
11   "value": [
12     {
13
14       "Id": "a9db1234-16f6-45c2-bb1b-a6f978fe9810",
15       "Sid": "S-1-5-21-2061182105-2775619996-3779015820-12345",
16       "Name": "EWS\\LR0003",
17       "IPAddress": null,
18       "IsAssigned": true,
19       "IsInMaintenanceMode": false,
20       "LastDeregisteredCode": 2,
21       "CurrentPowerState": 2,
22       "CurrentSessionCount": 0,
23       "FailureDate": null,
24       "FaultState": 1,
25       "LastPowerActionFailureReason": 401,
26       "LastPowerActionCompletedDate": "2025-05-13T04:59:50.18Z",
27       "DesktopGroup": {
28
29         "Name": "LR0-2-P"
30       }
31     }
32   ]
33 }
34
35 }
```

Alive sessions on each machine

- `Sessions` in the `$expand` operation enriches the response records with the sessions running on a machine.
- `EndDate eq null` represents the sessions that are still alive.

```
1 GET /monitorodata/Machines?$select=Name,IsInMaintenanceMode,
      CurrentRegistrationState,CurrentSessionCount,IsPreparing,FaultState&
      $expand=DesktopGroup($select=Id,Name),Sessions($select=SessionKey,
      ConnectionState;$filter=EndDate+eq+null)&$filter=Name+ne+null+and+
      DesktopGroup+ne+null HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
```

```

4 Content-Length: 122546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
      Machines",
11   "value": [
12     {
13
14       "Name": "ENG\\ra-s19-cvad079",
15       "IsInMaintenanceMode": false,
16       "CurrentRegistrationState": 1,
17       "CurrentSessionCount": 1,
18       "IsPreparing": false,
19       "FaultState": 1,
20       "DesktopGroup": {
21
22         "Id": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",
23         "Name": "CVD\\BUR CVAD Development"
24       }
25     },
26     "Sessions": [
27       {
28
29         "SessionKey": "d0d8b5ab-9450-4beb-9f3e-9f7c1577d9bd",
30         "ConnectionState": 2
31       }
32     ]
33   ]
34 }
35
36 ]
37 }

```

Active VDA machines associated information

- `LifecycleState` eq 0 represents active machines.
- `MachineRole` eq 0 represents a VDA machine.
- Use `$expand` to get associated information like catalog, hypervisor and load.

Request

```

1 GET /monitorodata/Machines?$filter=LifecycleState eq 0 and Name ne null
      and DnsName ne '' and CatalogId ne null and MachineRole eq 0&
      $expand=CurrentLoadIndex,Catalog,Hypervisor HTTP/1.1
2 Host: api.cloud.com

```

```
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 122546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11         Machines",
12     "value": [
13         {
14             "Id": "17f8b8f5-0f27-4852-a399-00194801a283",
15             "Sid": "S-1-5-21-1924580635-3864722089-3907235813-107439",
16             "Name": "ENG\\ra-s19-cvad036",
17             "DnsName": "ra-s19-cvad036.eng.citrite.net",
18             "LifecycleState": 0,
19             "IPAddress": null,
20             "HostedMachineId": "ccf63697-723c-0845-17fe-ebe1e2e75d31",
21             "HostingServerName": "ra-xen-07",
22             "HostedMachineName": "ra-s19-cvad036",
23             "IsAssigned": false,
24             "IsInMaintenanceMode": false,
25             "IsPendingUpdate": false,
26             "AgentVersion": "2212.0.0.77",
27             "AssociatedUserFullNames": "",
28             "AssociatedUserNames": "",
29             "AssociatedUserUPNs": "",
30             "CurrentRegistrationState": 2,
31             "RegistrationStateChangeDate": "2024-08-05T22:03:08.31Z",
32             "LastDeregisteredCode": 0,
33             "LastDeregisteredDate": "2024-08-05T22:02:33.64Z",
34             "CurrentPowerState": 2,
35             "CurrentSessionCount": 0,
36             "ControllerDnsName": "A21828-46-2.prodcp5.local",
37             "PoweredOnDate": "2024-08-05T10:01:50.4Z",
38             "PowerStateChangeDate": "2024-08-05T22:03:08.31Z",
39             "FunctionalLevel": 10,
40             "FailureDate": null,
41             "WindowsConnectionSetting": 0,
42             "IsPreparing": false,
43             "FaultState": 1,
44             "OSType": "Windows 2019",
45             "CurrentLoadIndexId": 17863392,
```

```
46     "CatalogId": "11c5298e-4516-4c1b-b83a-efb219009b71",
47     "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",
48     "HypervisorId": "56ac3d9f-2691-4d2c-aacb-95726611fe79",
49     "LastPowerActionType": 3,
50     "LastPowerActionReason": 8,
51     "LastPowerActionFailureReason": 400,
52     "LastPowerActionCompletedDate": "2024-08-05T22:02:50.887Z",
53     "Tags": [],
54     "Hash": "",
55     "MachineRole": 0,
56     "CreatedDate": "2020-06-10T14:31:36.183Z",
57     "ModifiedDate": "2024-08-05T22:47:00.103Z",
58     "CurrentLoadIndex": {
59
60         "Id": 17863392,
61         "EffectiveLoadIndex": 0,
62         "Cpu": 0,
63         "Memory": 0,
64         "Disk": 0,
65         "Network": 0,
66         "SessionCount": 0,
67         "MachineId": "17f8b8f5-0f27-4852-a399-00194801a283",
68         "CreatedDate": "2024-08-05T22:03:12.277Z",
69         "ModifiedDate": "2024-08-06T06:51:36.9277153Z"
70     },
71     ,
72     "Catalog": {
73
74         "Id": "11c5298e-4516-4c1b-b83a-efb219009b71",
75         "Name": "CVD\\BUR CVAD Development",
76         "LifecycleState": 0,
77         "ProvisioningType": 2,
78         "PersistentUserChanges": 1,
79         "IsMachinePhysical": false,
80         "AllocationType": 2,
81         "SessionSupport": 1,
82         "ProvisioningSchemeId": null,
83         "CreatedDate": "2020-05-19T00:02:03.903Z",
84         "ModifiedDate": "2022-12-10T13:51:24.173Z"
85     },
86     ,
87     "Hypervisor": {
88
89         "Id": "56ac3d9f-2691-4d2c-aacb-95726611fe79",
90         "Name": "BUR_Prod0",
91         "Type": "XenServer",
92         "LifecycleState": 0,
93         "CreatedDate": "2020-05-15T12:32:30.587Z",
94         "ModifiedDate": "2024-03-02T00:07:05.303Z"
95     },
96
97     }
98
```

```
99     ]
100  }
```

Get VDA machine availability over time

- `MachineSummaries` entity provides aggregate machine metrics at hourly or daily granularity.
- The `$expand` operations enriches the records with Desktop Group information.

Request

```
1 GET /monitorodata/MachineSummaries?$filter=Granularity eq 60 and
    SummaryDate ge 2025-05-01&$select=SummaryDate,PoweredOnMachinesCount
    ,RegisteredMachinesCount,MachinesInMaintenanceModeCount,
    MachinesCount&$expand=DesktopGroup($select=Id,Name)&$orderby=
    DesktopGroup/Name HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 2546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Machines",
11     "value": [
12         {
13
14             "SummaryDate": "2025-05-01T00:00:00Z",
15             "PoweredOnMachinesCount": 7,
16             "RegisteredMachinesCount": 5,
17             "MachinesInMaintenanceModeCount": 0,
18             "MachinesCount": 8,
19             "DesktopGroup": {
20
21                 "Id": "002683cd-b3a5-4fc8-913c-9f0d12e4d7cb",
22                 "Name": "LAB3-W10-P"
23             }
24         }
25     ]
26 }
```

```
25     }
26   ,
27     {
28
29       "SummaryDate":"2025-05-01T01:00:00Z",
30       "PoweredOnMachinesCount":0,
31       "RegisteredMachinesCount":0,
32       "MachinesInMaintenanceModeCount":1,
33       "MachinesCount":2,
34       "DesktopGroup":{
35
36         "Id":"002683cd-b3a5-4fc8-913c-9f0d12e4d7cb",
37         "Name":"LAB3-W10-P"
38       }
39     }
40   }
41 ]
42 ]
43 }
```

List of Hotfixes applied to each machine

As part of this example, we query the MachineHotfixLogs entity, which includes records with each of the hotfixes applied to the VDA machines. We enrich the result records with Machine and Desktop-Group information.

- `MachineHotfixLogs` provides a list of the hotfixes.
- Use `$expand` to get associated Machine, DesktopGroup and Hotfix information.

Request

```
1 GET /monitorodata/MachineHotfixLogs?$filter=CreatedDate ge 2025-02-01&
   $expand=Machine($select=Name;$expand=DesktopGroup($select=Name)),
   Hotfix&$orderby=Machine/DesktopGroup/Name,Machine/Name HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 2546
5 Connection: keep-alive
6 odata-version: 4.0
```

```

7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
      Machines(CurrentLoadIndex(),Catalog(),Hypervisor())",
11   "value": [
12     {
13
14       "Id":10073,
15       "MachineId":"6829529e-4126-417c-90a7-a372dc7bc7",
16       "HotfixId":"33fd700d-d35f-4b6c-81e0-30cd793547f7",
17       "ChangeType":1,
18       "CurrentState":true,
19       "CreatedDate":"2024-11-27T13:24:38.05Z",
20       "ModifiedDate":"2024-11-27T13:24:38.05Z",
21       "Machine":{
22
23         "Name":"LAB\\TEST-MVDA-01",
24         "DesktopGroup":{
25
26           "Name":"DG-W10-PMD"
27         }
28       }
29     },
30     "Hotfix":{
31
32       "Id":"33fd700d-d35f-4b6c-81e0-30cd793547f7",
33       "Name":"IdentityAssertionWX64_7_33_5000",
34       "Article":"http://support.citrix.com/article/CTX579943"
35     },
36     "ArticleName":"CTX579943",
37     "FileName":"Citrix.Authentication.IdentityAssertion_x64",
38     "FileFormat":"MSI",
39     "Version":"10.9.5000.8",
40     "ComponentName":"Citrix Identity Assertion VDA Plugin",
41     "ComponentVersion":"10.9.5000.8",
42     "CreatedDate":"2024-06-07T19:42:49.38Z",
43     "ModifiedDate":"2024-06-07T19:42:49.38Z"
44   }
45 }
46 ]
47
48 }
49

```

Get a list of all the VDA versions

Query the Machines entity, aggregate over the distinct versions and calculate the number of machines running each of the VDA versions per delivery group.

Request

```
1 GET /monitorodata/Machines?$apply=filter(AgentVersion ne null and
   DesktopGroup/Name ne null)/expand(DesktopGroup,filter(Name ne null))
   /groupby((DesktopGroup/Name,AgentVersion),aggregate(Id with
   countdistinct as MachineCount)) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 08:00:05 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 2546
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Machines(CurrentLoadIndex(),Catalog(),Hypervisor())",
11   "value": [
12     {
13
14       "@odata.id": null,
15       "AgentVersion": "2407.0.10.101",
16       "MachineCount": 46,
17       "DesktopGroup": {
18
19         "@odata.id": null,
20         "Name": "DG-W10-LRO-P"
21       }
22     }
23   ,
24     {
25
26       "@odata.id": null,
27       "AgentVersion": "2402.0.100.609",
28       "MachineCount": 11,
29       "DesktopGroup": {
30
31         "@odata.id": null,
32         "Name": "DG-W10-PMD"
33       }
34     }
35   }
36 ]
37
38 ]
```

Machine resource utilization request examples

June 2, 2025

Use Citrix Monitor Service API to get machine level CPU and memory usage.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/ResourceUtilizationSummary` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

Powered on machine CPU and memory usage at hourly level

- `Granularity = 60` represents hourly level metrics.
- `UptimeInMinutes > 0` represents the machine powered on in this hour.

Request

```
1 GET /monitorodata/ResourceUtilizationSummary?$filter=Granularity eq 60
   and SummaryDate eq 2024-07-30T08:00:00Z and UptimeInMinutes gt 0&
   $expand=Machine($select=Name,AgentVersion),DesktopGroup($select=Name
   ) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     ResourceUtilizationSummary",
12     "value": [
13         {
14             "MachineId": "31a02fb0-b673-4520-b94d-017fa2acd3b8",
15             "Granularity": 60,
16             "SummaryDate": "2025-05-05T08:00:00Z",
17             "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",
18             "AvgPercentCpu": 0.99298245335618651,
19             "PeakPercentCpu": 3.5789473056793213,
20             "AvgUsedMemory": 7918633224.9824562,
21             "PeakUsedMemory": 7937829187.3684206,
22             "AvgTotalMemory": 17170366464.0,
23             "TotalSessionCount": 1,
24             "AvgIcaRttInMs": null,
25             "AvgLogOnDurationInMs": null,
26             "UptimeInMinutes": 60,
27             "UpTimeWithoutSessionInMinutes": 0,
28             "DisconnectTimeInMinutes": 60,
29             "CreatedDate": "2024-07-30T09:13:46.247Z",
30             "ModifiedDate": "2024-07-30T09:13:46.247Z",
31             "Machine": {
32
33                 "Name": "EWS\\MI-PMD005",
34                 "AgentVersion": "2407.0.100.274"
35             }
36         },
37         "DesktopGroup": {
38
39             "Name": "MI-PMD"
40         }
41     }
42 ]
43
44 }
45 }
```

Powered on machine CPU and memory usage at day level

- `Granularity = 1440` represents day level metrics.
- `UptimeInMinutes > 0` represents the machine powered on in this day.

Request

```
1 GET /monitorodata/ResourceUtilizationSummary?$filter=Granularity eq
  1440 and SummaryDate eq 2024-07-30T00:00:00Z and UptimeInMinutes gt
  0 HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
  ResourceUtilizationSummary",
11   "value": [
12     {
13
14       "MachineId": "17f8b8f5-0f27-4852-a399-00194801a283",
15       "Granularity": 1440,
16       "SummaryDate": "2025-05-05T00:00:00Z",
17       "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",
18       "AvgPercentCpu": 1.1274837264250215,
19       "PeakPercentCpu": 14.733333587646484,
20       "AvgUsedMemory": 4474693852.6077623,
21       "PeakUsedMemory": 4756543078.4,
22       "AvgTotalMemory": 17170366464.0,
23       "TotalSessionCount": 0,
24       "AvgIcaRttInMs": null,
25       "AvgLogOnDurationInMs": null,
26       "UptimeInMinutes": 724,
27       "UpTimeWithoutSessionInMinutes": null,
28       "DisconnectTimeInMinutes": null,
29       "IdleTimeInMinutes": null,
30       "CreatedDate": "2024-07-31T00:13:46.027Z",
31       "ModifiedDate": "2024-07-31T00:13:46.027Z",
32       "Machine": {
```

```
33
34         "Name": "EWS\\MI-PMD005",
35         "AgentVersion": "2407.0.100.274"
36     }
37 ,
38     "DesktopGroup": {
39         "Name": "MI-PMD"
40     }
41 }
42
43 }
44
45 ]
46 }
```

Machine resource utilization aggregates by delivery group

The query is more complex, as we first need to compute percentage values based on the used and total memory metrics. We group by the DesktopGroup name and the hourly timestamp and calculate average and max values.

- `$apply` operation includes a `filter` stage, a `compute` stage, an `expand` stage and a `groupby/aggregate` stage. See the [OData documentation](#) for usage details.

Request

```
1 GET /monitorodata/ResourceUtilizationSummary?$apply=filter(Granularity
  eq 60 and SummaryDate ge 2025-05-03 and UptimeInMinutes gt 0 and
  AvgTotalMemory gt 0 and AvgUsedMemory gt 0 and PeakUsedMemory gt 0)/
  compute(100 mul AvgUsedMemory div AvgTotalMemory as AvgPercentMemory
  , 100 mul PeakUsedMemory div AvgTotalMemory as PeakPercentMemory)/
  expand(DesktopGroup,filter(Name ne null))/groupby((DesktopGroup/Name
  ,SummaryDate),aggregate(AvgPercentCpu with average as AvgPercentCpu,
  PeakPercentCpu with max as PeakPercentCpu, AvgPercentMemory with
  average as AvgPercentMemory, PeakPercentMemory with max as
  PeakPercentMemory)) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
```

```
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11         ResourceUtilizationSummary",
12     "value": [
13         {
14             "@odata.id": null,
15             "SummaryDate": "2025-05-03T00:00:00Z",
16             "PeakPercentMemory": 14.249562365700822,
17             "AvgPercentMemory": 14.176066487959879,
18             "PeakPercentCpu": 1.7819637060165405,
19             "AvgPercentCpu": 0.64738541841506958,
20             "DesktopGroup": {
21
22                 "@odata.id": null,
23                 "Name": "MI-PMD"
24             }
25         },
26         {
27
28             "@odata.id": null,
29             "SummaryDate": "2025-05-03T01:00:00Z",
30             "PeakPercentMemory": 14.241976541431665,
31             "AvgPercentMemory": 14.18313095499029,
32             "PeakPercentCpu": 2.2718720436096191,
33             "AvgPercentCpu": 0.65315361817677819,
34             "DesktopGroup": {
35
36                 "@odata.id": null,
37                 "Name": "MI-PMD"
38             }
39         }
40     ]
41 }
42
43
44 ]
45 }
```

Daily average uptime by delivery group

The query gets the daily resource utilization summaries and averages the UptimeInMinutes by Desktop Group over the defined time period.

- `$apply` operation includes a `filter` stage, an `expand` stage and a `groupby/aggregate` stage. See the [OData documentation](#) for usage details.

Request

```
1 GET /monitorodata/ResourceUtilizationSummary?$apply=filter(Granularity
   eq 1440 and SummaryDate ge 2025-05-01 and SummaryDate le 2025-05-10)
   /expand(Machine,expand(DesktopGroup,filter(Name ne null)))/groupby((
   DesktopGroupId, DesktopGroup/Name), aggregate(cast(UptimeInMinutes,
   Edm.Int32) with average as AvgUptimeInMinutes)) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   ResourceUtilizationSummary",
11   "value": [
12     {
13
14       "@odata.id": null,
15       "DesktopGroupId": "97123b25-ad22-4a37-a6bd-0355952fdfe1",
16       "AvgUptimeInMinutes": 1222.7,
17       "DesktopGroup": {
18
19         "@odata.id": null,
20         "Name": "LRO-P"
21       }
22     }
23   ],
24   {
25     {
26
27       "@odata.id": null,
28       "DesktopGroupId": "e3123b36-6807-490c-b71e-2af99bbfb0d8",
29       "AvgUptimeInMinutes": 1437.75,
30       "DesktopGroup": {
31
32         "@odata.id": null,
33         "Name": "MacVDA"
34       }
35     }
36   }
37 ,
```

```
38     {
39
40         "@odata.id": null,
41         "DesktopGroupId": "4e123ac2-99d8-49a9-a082-3949284b0e8d",
42         "AvgUptimeInMinutes": 993.9785714285714,
43         "DesktopGroup": {
44
45             "@odata.id": null,
46             "Name": "W10-TS-P"
47         }
48     }
49 }
50
51 ]
52 }
```

Unused machines over time period

In this example, we calculate the total number of sessions running on each of the machines over the selected time period and then filter for the machines with no sessions. Note that we have to cast the TotalSessionCount to a larger integer type (is Int16) for the aggregation to succeed.

Request

```
1 GET /monitorodata/ResourceUtilizationSummary?$apply=filter(Granularity
   eq 60 and SummaryDate ge 2025-05-05 and UptimeInMinutes gt 0)/expand
   (Machine,expand(DesktopGroup,filter(Name ne null)))/compute(cast(
   TotalSessionCount,Edm.Int64) as TotalSessionCountInt64 )/groupby((
   DesktopGroup/Name,MachineId,Machine/Name),aggregate(
   TotalSessionCountInt64 with sum as TotalSessionCountSum))/filter(
   TotalSessionCountSum eq 0)&$count=true HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
```

```
10  "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    ResourceUtilizationSummary",
11  "value": [
12    {
13
14      "@odata.id":null,
15      "MachineId":"5bf5d243-2395-46d1-bd7e-83f77ae65db3",
16      "TotalSessionCountSum":0,
17      "Machine": {
18
19        "@odata.id": null,
20        "Name": "LAB\\TEST-MVDA-01"
21      }
22    },
23    "DesktopGroup":{
24
25      "@odata.id":null,
26      "Name":"DG-W10-PMD"
27    }
28  }
29  },
30  {
31    {
32
33      "@odata.id":null,
34      "MachineId":"a9dbe063-16f6-45c2-bb1b-a6f978fe9810",
35      "TotalSessionCountSum":0,
36      "Machine": {
37
38        "@odata.id": null,
39        "Name": "LAB\\TEST-MVDA-02"
40      }
41    },
42    "DesktopGroup":{
43
44      "@odata.id":null,
45      "Name":"DG-W10-LR0-2-P"
46    }
47  }
48  }
49  ]
50  }
51  }
```

Session data request examples

June 2, 2025

The Session entity represents a user connected to a desktop or an application.

Use Citrix Monitor Service API to get the data for entity session.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/Sessions` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

ApplicationInstances for active sessions

- `ConnectionState eq 5` represents the sessions with active connection state. Please reference [ConnectionState](#) for a description of each connection state value.
- `ModifiedDate` represents the target time range.

Request

```
1 GET /monitorodata/Sessions?$expand=ApplicationInstances&$filter=(
    ConnectionState eq 5) and (ModifiedDate gt 2024-08-06T07:00:00Z and
    ModifiedDate le 2024-08-06T07:10:00Z ) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
```

```

10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
      Sessions(ApplicationInstances())",
11     "value": [
12         {
13
14             "SessionKey": "4569fdfd-54c4-44c8-81a9-40688a2771d6",
15             "StartDate": "2024-07-31T11:22:37.473Z",
16             "LogOnDuration": null,
17             "EndDate": null,
18             "ExitCode": 1,
19             "FailureId": null,
20             "FailureDate": null,
21             "ConnectionState": 5,
22             "SessionIdleTime": null,
23             "ConnectionStateChangeDate": "2024-08-06T07:08:50.72Z",
24             "LifecycleState": 0,
25             "CurrentConnectionId": 887132,
26             "UserId": 6,
27             "MachineId": "7eb414f9-533c-498f-a72f-b4459625100e",
28             "SessionType": 0,
29             "IsAnonymous": false,
30             "PublishedDesktopId": 93,
31             "CreatedDate": "2024-07-31T11:22:37.473Z",
32             "ModifiedDate": "2024-08-06T07:08:50.72Z",
33             "ApplicationInstances": []
34         }
35     ]
36 }
37 }

```

Connection and machine information of each session

- Connection represents an initial connection or reconnect for a session.
- Use `$expand` operation to get the connection state for each session.

Request

```

1 GET /monitorodata/Sessions?$select=StartDate,LogOnDuration,EndDate&
  $expand=Machine($select=DesktopGroupId),Connections($select=
  LogOnStartDate,LogOnEndDate,BrokeringDuration,VMStartStartDate,
  VMStartEndDate,HdxStartDate,HdxEndDate,AuthenticationDuration,
  GpoStartDate,GpoEndDate,LogOnScriptsStartDate,LogOnScriptsEndDate,
  ProfileLoadStartDate,ProfileLoadEndDate,InteractiveStartDate,
  InteractiveEndDate;$filter=IsReconnect+eq+false)&$filter=
  LogOnDuration+ne+null+and+StartDate+gt+cast(2024-07-31T02:56:57.996Z
  ,+Edm.DateTimeOffset)&$orderby=StartDate+desc HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Sessions(StartDate,LogOnDuration,EndDate,Machine(DesktopGroupId)
    ,Connections(LogOnStartDate,LogOnEndDate,BrokeringDuration,
    VMStartStartDate,VMStartEndDate,HdxStartDate,HdxEndDate,
    AuthenticationDuration,GpoStartDate,GpoEndDate,
    LogOnScriptsStartDate,LogOnScriptsEndDate,ProfileLoadStartDate,
    ProfileLoadEndDate,InteractiveStartDate,InteractiveEndDate))",
11   "value": [
12     {
13
14       "StartDate": "2024-07-31T19:53:41.913Z",
15       "LogOnDuration": 147061,
16       "EndDate": null,
17       "Machine": {
18
19         "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596
20         "
21       }
22     },
23     "Connections": [
24       {
25
26         "LogOnStartDate": "2024-07-31T19:53:41.913Z",
27         "LogOnEndDate": "2024-07-31T19:55:59.583Z",
28         "BrokeringDuration": 34,
29         "VMStartStartDate": null,
30         "VMStartEndDate": null,
31         "HdxStartDate": null,
32         "HdxEndDate": null,
33         "AuthenticationDuration": 266,
34         "GpoStartDate": "2024-07-31T19:55:30.857Z",
35         "GpoEndDate": "2024-07-31T19:55:57.037Z",
36         "LogOnScriptsStartDate": null,
37         "LogOnScriptsEndDate": null,
38         "ProfileLoadStartDate": "2024-07-31T19:55:28.13Z",
39         "ProfileLoadEndDate": "2024-07-31T19:55:30.853Z",
40         "InteractiveStartDate": "2024-07-31T19:55:30.853Z",
41         "InteractiveEndDate": "2024-07-31T19:56:09.01Z"
42       }
43     ]
44   }
}
```

```

45
46     ]
47 }

```

SessionMetrics for alive sessions

- `EndDate` is `null` represents alive sessions.
- Use `$expand=SessionMetrics` to get the metrics of each session.

Request

```

1 GET /monitorodata/Sessions?$filter=EndDate eq null &$select=StartDate,
    ConnectionState,SessionIdleTime&$expand=SessionMetrics($select=
    IcaRttMS),Machine($expand=DesktopGroup($select=Name)),Machine(
    $select=Name),User($select=UserName),CurrentConnection($select=
    ClientName,ClientVersion,ClientAddress,ConnectedViaIPAddress,
    BrokeringDuration) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Sessions(StartDate,ConnectionState,SessionIdleTime,
    SessionMetrics(IcaRttMS),Machine(Name,DesktopGroup(Name)),User(
    UserName),CurrentConnection(ClientName,ClientVersion,
    ClientAddress,ConnectedViaIPAddress,BrokeringDuration))",
11     "value": [
12         {
13
14             "StartDate": "2024-06-25T07:46:56.673Z",
15             "ConnectionState": 2,
16             "SessionIdleTime": null,
17             "SessionMetrics": [
18                 {
19
20                     "IcaRttMS": 0

```

```
21         }
22     ,
23         {
24         "IcaRttMS": 0
25         }
26     ],
27     "Machine": {
28         "Name": "ENG\\ra-s19-cvad025",
29         "DesktopGroup": {
30             "Name": "CVD\\BUR CVAD Development"
31         }
32     }
33     ,
34     "User": {
35         "UserName": "ameyag"
36     }
37     ,
38     "CurrentConnection": {
39         "ClientName": "AMERENGAAD62201",
40         "ClientAddress": "10.122.198.42",
41         "ClientVersion": "24.5.0.131",
42         "ConnectedViaIPAddress": "3.3.11.7",
43         "BrokeringDuration": null
44     }
45     }
46     ,
47     {
48         "StartDate": "2024-07-26T20:35:12.673Z",
49         "ConnectionState": 2,
50         "SessionIdleTime": "2024-07-26T20:48:00Z",
51         "SessionMetrics": [
52             {
53                 "IcaRttMS": 0
54             }
55             ,
56             {
57                 "IcaRttMS": 0
58             }
59         ]
60     }
61     ],
62     "Machine": {
```

```
74         "Name": "ENG\\SHADOWS",
75         "DesktopGroup": {
76             "Name": "Remote PC - London"
77         }
78     },
79     "User": {
80         "UserName": "paulbro"
81     },
82     "CurrentConnection": {
83         "ClientName": "PAULMACHINE",
84         "ClientAddress": "192.168.68.88",
85         "ClientVersion": "24.5.0.131",
86         "ConnectedViaIPAddress": "10.3.72.60",
87         "BrokeringDuration": null
88     }
89 ]
90 }
```

Alive sessions for target DesktopGroup

- `Machine/DesktopGroup/Id eq ad9e2d3d-44ac-49b0-a34b-7affb5ce3596` represents the target Desktop Group.
- `EndDate eq null` represent alive sessions.

Request

```
1 GET /monitorodata/Sessions?$expand=Machine($expand=DesktopGroup)&
   $filter=(Machine/DesktopGroup/Id eq ad9e2d3d-44ac-49b0-a34b-7
   affb5ce3596) and (EndDate eq null) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
```

```
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
        Sessions(Machine(DesktopGroup()))",
11     "value": [
12         {
13
14             "SessionKey": "2d53d7f8-477c-4b18-a060-0081e050cc25",
15             "StartDate": "2024-06-25T07:46:56.673Z",
16             "LogOnDuration": 122030,
17             "EndDate": null,
18             "ExitCode": 1,
19             "FailureId": null,
20             "FailureDate": null,
21             "ConnectionState": 2,
22             "SessionIdleTime": null,
23             "ConnectionStateChangeDate": "2024-07-31T21:22:44.527Z",
24             "LifecycleState": 0,
25             "CurrentConnectionId": 886035,
26             "UserId": 4320,
27             "MachineId": "43a6ac00-9263-4c1b-acf5-f68267c6a26f",
28             "SessionType": 0,
29             "IsAnonymous": false,
30             "PublishedDesktopId": 3,
31             "CreatedDate": "2024-06-25T07:48:43.95Z",
32             "ModifiedDate": "2024-07-31T21:22:44.527Z",
33             "Machine": {
34
35                 "Id": "43a6ac00-9263-4c1b-acf5-f68267c6a26f",
36                 "Sid": "S
                    -1-5-21-1924580635-3864722089-3907235813-171780",
37                 "Name": "ENG\\ra-s19-cvad078",
38                 "DnsName": "ra-s19-cvad078.eng.citrite.net",
39                 "LifecycleState": 0,
40                 "IPAddress": "10.60.8.178",
41                 "HostedMachineId": "e59ef480-56b7-8f02-5233-
                    efd5a161cc41",
42                 "HostingServerName": "ra-xen-02",
43                 "HostedMachineName": "ra-s19-cvad078",
44                 "IsAssigned": false,
45                 "IsInMaintenanceMode": false,
46                 "IsPendingUpdate": false,
47                 "AgentVersion": "2212.0.0.77",
48                 "AssociatedUserFullNames": "Nataliia Taras",
49                 "AssociatedUserNames": "CITRITE\\t_nataliiat",
50                 "AssociatedUserUPNs": "Nataliia.Taras@citrix.com",
51                 "CurrentRegistrationState": 1,
52                 "RegistrationStateChangeDate": "2024-07-05T04:07:06.44Z
                    ",
```

```
53     "LastDeregisteredCode": 0,  
54     "LastDeregisteredDate": "2024-06-24T05:33:40.443Z",  
55     "CurrentPowerState": 3,  
56     "CurrentSessionCount": 1,  
57     "ControllerDnsName": "A21828-44-2.prodcp5.local",  
58     "PoweredOnDate": "2024-06-24T10:06:16.1Z",  
59     "PowerStateChangeDate": "2024-07-12T18:50:38.627Z",  
60     "FunctionalLevel": 10,  
61     "FailureDate": null,  
62     "WindowsConnectionSetting": 0,  
63     "IsPreparing": false,  
64     "FaultState": 1,  
65     "OSType": "Windows 2019",  
66     "CurrentLoadIndexId": 17861538,  
67     "CatalogId": "11c5298e-4516-4c1b-b83a-efb219009b71",  
68     "DesktopGroupId": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",  
69     "HypervisorId": "56ac3d9f-2691-4d2c-aacb-95726611fe79",  
70     "LastPowerActionType": 1,  
71     "LastPowerActionReason": 8,  
72     "LastPowerActionFailureReason": 400,  
73     "LastPowerActionCompletedDate": "2024-06-24T10  
74         :06:21.937Z",  
75     "Tags": [],  
76     "Hash": "",  
77     "MachineRole": 0,  
78     "CreatedDate": "2021-04-27T12:41:12.097Z",  
79     "ModifiedDate": "2024-07-12T18:50:38.627Z",  
80     "DesktopGroup": {  
81         "Id": "ad9e2d3d-44ac-49b0-a34b-7affb5ce3596",  
82         "Name": "CVD\\BUR CVAD Development",  
83         "IsRemotePC": false,  
84         "DesktopKind": 1,  
85         "LifecycleState": 0,  
86         "SessionSupport": 1,  
87         "DeliveryType": 2,  
88         "IsInMaintenanceMode": false,  
89         "MachineCost": 0.0,  
90         "CurrencyType": null,  
91         "AutoscaleTagId": 0,  
92         "CreatedDate": "2020-05-19T00:07:37.387Z",  
93         "ModifiedDate": "2024-06-04T00:05:01.13Z"  
94     }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 ]  
101 }
```

Terminated sessions in a time range

- `ConnectionState` ne 3 represents terminated sessions.
- `FailureDate` gt 2020-12-04T15:08:25.000Z gets failures after the mentioned time-stamp.

Request

```
1 GET /monitorodata/Sessions?$filter=ConnectionState ne 3 or FailureDate
   gt 2020-12-04T15:08:25.000Z HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Sessions",
11     "value": [
12         {
13
14             "SessionKey": "3f570c39-a0d0-4dda-9e2e-001040b22ea7",
15             "StartDate": "2024-05-17T09:22:06.87Z",
16             "LogOnDuration": null,
17             "EndDate": "2024-05-17T09:22:06.87Z",
18             "ExitCode": 101,
19             "FailureId": 13,
20             "FailureDate": "2024-05-17T09:22:06.87Z",
21             "ConnectionState": 3,
22             "SessionIdleTime": null,
23             "ConnectionStateChangeDate": "2024-05-17T09:22:06.87Z",
24             "LifecycleState": 3,
25             "CurrentConnectionId": 857484,
26             "UserId": 2762,
27             "MachineId": "510ff5f6-887c-4a43-a042-c6bf03313da0",
28             "SessionType": 0,
29             "IsAnonymous": false,
30             "PublishedDesktopId": null,
31             "CreatedDate": "2024-05-17T09:22:06.87Z",
```

```
32         "ModifiedDate": "2024-05-17T09:22:06.87Z"
33     }
34
35 ]
36 }
```

Running Desktop sessions

- `ConnectionState eq 5` and `EndDate eq null` represents running sessions.
- `SessionType eq 0` represents desktop sessions

Request

```
1 GET /monitorodata/Sessions?$filter=ConnectionState eq 5 and EndDate eq
  null and SessionType eq 0&$select=SessionKey,StartDate,SessionType&
  $expand=PublishedDesktopName,User($select=Fullname,Upn),Machine(
  $select=Name)&$count=true HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
  Sessions",
11     "value": [
12         {
13
14             "SessionKey": "1b8b2cbf-8281-49a5-9822-00026b2ef1f8",
15             "StartDate": "2024-11-26T08:18:36.743Z",
16             "SessionType": 0,
17             "PublishedDesktopName": {
18
19                 "Id": 5,
20                 "PublishedName": "Desktop-W10-TS-P"
21             }
22         },
23         "User": {
```

```
24
25         "Upn":"user.name@citrix.com",
26         "FullName":"User Name"
27     }
28 ,
29     "Machine":{
30
31         "Name":"LAB\\TEST-MVDA-01"
32     }
33
34 }
35
36 ]
37 }
```

Number of active sessions by Desktop Group and Session Type

As part of this example, we query the Sessions entity, but expand to the DesktopGroup via the Machine entity. We group by the DesktopGroup name and the Session type. The list of Session Types can be found here: [SessionType](#).

Request

```
1 GET /monitorodata/Sessions?$apply=filter(StartDate ge 2025-05-02 and
   EndDate eq null)/expand(Machine,expand(DesktopGroup,filter(Name ne
   null)))/groupby((Machine/DesktopGroup/Name,SessionType),aggregate(
   SessionKey with countdistinct as SessionCount)) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Sessions",
11     "value": [
12         {
13
```

```
14     "@odata.id":null,
15     "SessionType":0,
16     "SessionCount":36,
17     "Machine":{
18
19         "@odata.id":null,
20         "DesktopGroup":{
21
22             "@odata.id":null,
23             "Name":"LAB3-W10-P"
24         }
25     }
26 }
27 }
28 }
29 ]
30 ]
31 }
```

Connection data request examples

February 9, 2026

Connection represents an initial connection or reconnect for a session.

Use Citrix Monitor Service API to get the details of connection.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/Connections` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

Connection details for sessions with failure

- Use `$expand` operation to get the details for sessions with failure.

Request

```
1 /monitorodata/Connections?$filter=Session/FailureDate ge 2024-07-24T00
  :00:00.000Z and Session/FailureDate le 2024-07-25T00:00:00.000Z &
  $select=*,&$expand=Session($select=*$;$expand=User($select=UPN,
  UserName),Failure($select=ConnectionFailureEnumValue,Category),
  Machine($select=DnsName,DesktopGroupId,IsInMaintenanceMode)) HTTP
  /1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8
9 {
10
11   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Connections(*,Session(*,User(Upn,UserName),Failure(
    ConnectionFailureEnumValue,Category),Machine(DnsName,
    DesktopGroupId,IsInMaintenanceMode)))",
12   "value": [
13     {
14
15       "Id": 408601,
16       "ClientName": "test_s_MacBook_Pr",
17       "ClientAddress": "192.168.50.144",
18       "ClientVersion": "22.01.0.9",
19       "ClientPlatform": "Macintosh",
20       "ConnectedViaHostName": null,
21       "ConnectedViaIPAddress": "10.69.78.36",
22       "LaunchedViaHostName": "Workspace",
23       "LaunchedViaIPAddress": null,
24       "IsReconnect": false,
25       "IsSecureIca": false,
26       "Protocol": "HDX",
27       "LogOnStartDate": "2022-03-31T07:49:47.307Z",
28       "LogOnEndDate": "2022-03-31T07:50:00.33Z",
29       "BrokeringDuration": null,
30       "BrokeringDate": null,
31       "DisconnectCode": 0,
32       "DisconnectDate": "2022-03-31T08:10:51.533Z",
33       "VMStartStartDate": null,
34       "VMPoweredOnDate": null,
```

```
35     "VMStartEndDate": null,
36     "ClientSessionValidateDate": null,
37     "ServerSessionValidateDate": "2022-03-31T07:49:47.307Z",
38     "EstablishmentDate": "2022-03-31T07:49:47.307Z",
39     "HdxStartDate": null,
40     "HdxEndDate": null,
41     "AuthenticationDuration": null,
42     "GpoStartDate": null,
43     "GpoEndDate": null,
44     "LogOnScriptsStartDate": null,
45     "LogOnScriptsEndDate": null,
46     "ProfileLoadStartDate": null,
47     "ProfileLoadEndDate": null,
48     "InteractiveStartDate": null,
49     "InteractiveEndDate": null,
50     "SessionKey": "38502302-78ed-4f02-a3f0-cf6a31269b29",
51     "CreatedDate": "2022-03-31T07:50:00.33Z",
52     "ModifiedDate": "2022-03-31T08:10:51.533Z",
53     "Session": {
54
55         "SessionKey": "38502302-78ed-4f02-a3f0-cf6a31269b29",
56         "StartDate": "2022-03-31T07:49:47.307Z",
57         "LogOnDuration": null,
58         "EndDate": null,
59         "ExitCode": 1,
60         "FailureId": 13,
61         "FailureDate": "2024-07-24T14:53:56.517Z",
62         "ConnectionState": 2,
63         "SessionIdleTime": null,
64         "ConnectionStateChangeDate": "2022-04-22T07:47:14.63Z",
65         "LifecycleState": 0,
66         "CurrentConnectionId": 884385,
67         "UserId": 1505,
68         "MachineId": "7ff93079-914a-4e6e-b9fe-5ce6e7578684",
69         "SessionType": 0,
70         "IsAnonymous": false,
71         "PublishedDesktopId": null,
72         "CreatedDate": "2022-03-31T07:50:00.33Z",
73         "ModifiedDate": "2022-04-22T07:47:14.63Z",
74         "User": {
75
76             "Upn": "test.test@citrix.com",
77             "UserName": "test"
78         }
79     },
80     "Failure": {
81
82         "ConnectionFailureEnumValue": 101,
83         "Category": 2
84     }
85 },
86     "Machine": {
87
```

```
88         "DnsName": "test-office.eng.citrite.net",
89         "IsInMaintenanceMode": false,
90         "DesktopGroupId": "b62f4949-0756-4a1e-94d9-
          e70cdfd0a198"
91     }
92 }
93 }
94 }
95 }
96 ]
97 }
98 }
```

List of client versions

For the list of the client versions, we can query the Connections entity. We can aggregate over the Client platform and version and calculate the number of clients running each of the versions.

Request

```
1 /monitorodata/Connections?$apply=filter(ClientVersion ne null and
  ClientPlatform ne null)/groupby((ClientPlatform,ClientVersion),
  aggregate(ClientName with countdistinct as ClientCount)) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8
9 {
10
11   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Connections(*,Session(*,User(Upn,UserName),Failure(
    ConnectionFailureEnumValue,Category),Machine(DnsName,
    DesktopGroupId,IsInMaintenanceMode)))",
12   "value": [
13     {
14
15       "@odata.id": null,
```

```
16         "ClientVersion":"24.8.0",
17         "ClientPlatform":"Android",
18         "ClientCount":15
19     }
20     ,
21     {
22
23         "@odata.id":null,
24         "ClientVersion":"24.11.0.2065",
25         "ClientPlatform":"Windows",
26         "ClientCount":18
27     }
28 ]
29 }
30 }
```

Connection Failure log request examples

June 2, 2025

Use Citrix Monitor Service API to get the details of connection failures.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/ConnectionFailure` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

ConnectionFailureLogs for target time

- `$filter=(FailureDate ge 2024-07-30T00:00:00Z and FailureDate le 2024-07-31T00:00:00Z)` represents the target time range.

Request

```
1 GET /monitorodata/ConnectionFailureLogs?$filter=(FailureDate ge
    2024-07-30T00:00:00Z and FailureDate le 2024-07-31T00:00:00Z) HTTP
    /1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    ConnectionFailureLogs",
11     "value": [
12         {
13
14             "Id": 128230,
15             "SessionKey": "77337e10-1ad2-4e95-8990-b83a420405ed",
16             "FailureDate": "2024-07-30T01:02:49.37Z",
17             "UserId": 1225,
18             "MachineId": "53e6e3b6-a874-4e9b-ab4b-61241c34f241",
19             "ConnectionFailureEnumValue": 4,
20             "CreatedDate": "2024-07-30T01:02:49.37Z",
21             "ModifiedDate": "2024-07-30T01:02:51.387Z"
22         }
23     ,
24         {
25
26             "Id": 128231,
27             "SessionKey": "77337e10-1ad2-4e95-8990-b83a420405ed",
28             "FailureDate": "2024-07-30T01:19:49.98Z",
29             "UserId": 1225,
30             "MachineId": "53e6e3b6-a874-4e9b-ab4b-61241c34f241",
31             "ConnectionFailureEnumValue": 4,
32             "CreatedDate": "2024-07-30T01:19:49.98Z",
33             "ModifiedDate": "2024-07-30T01:19:52.727Z"
34         }
35     ,
36     ]
37 }
```

ConnectionFailureLogs related Machine, Session and User

- With `$expand` operation to get the related Machine, Session and User details for Connection-FailureLogs.

Request

```

1 GET /monitorodata/ConnectionFailureLogs?$filter=(FailureDate ge
    2024-07-31T00:00:00Z)&$expand=Machine($select=Name,
    CurrentRegistrationState,IsInMaintenanceMode),User($select=Sid,Upn,
    UserName,Domain),Session HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    ConnectionFailureLogs",
11     "value": [
12         {
13
14             "Id": 128249,
15             "SessionKey": "0b3e91c9-b324-4a32-871a-70cc8e9a7a2a",
16             "FailureDate": "2024-07-31T00:23:48.84Z",
17             "UserId": 1308,
18             "MachineId": "3563b00d-4c5b-4658-8fa2-9eddbf017e0b",
19             "ConnectionFailureEnumValue": 3,
20             "CreatedDate": "2024-07-31T00:23:48.84Z",
21             "ModifiedDate": "2024-07-31T00:23:51.547Z",
22             "Machine": {
23
24                 "Name": "ENG\\ra-w10-cvad010",
25                 "IsInMaintenanceMode": false,
26                 "CurrentRegistrationState": 2
27             }
28         },
29         "User": {
30

```

```
31     "Sid": "S
32         -1-5-21-1076320343-3137401123-2908695819-435603",
33     "Upn": "igor.polozkov@citrix.com",
34     "UserName": "igorp",
35     "Domain": "CITRITE"
36   }
37 ,
38   "Session": {
39     "SessionKey": "0b3e91c9-b324-4a32-871a-70cc8e9a7a2a",
40     "StartDate": "2024-07-19T13:30:02.003Z",
41     "LogOnDuration": 149790,
42     "EndDate": null,
43     "ExitCode": 1,
44     "FailureId": 11,
45     "FailureDate": "2024-07-31T13:38:20.69Z",
46     "ConnectionState": 2,
47     "SessionIdleTime": null,
48     "ConnectionStateChangeDate": "2024-07-25T13:57:20.283Z"
49   },
50   "LifecycleState": 0,
51   "CurrentConnectionId": 885929,
52   "UserId": 1308,
53   "MachineId": "3563b00d-4c5b-4658-8fa2-9eddbf017e0b",
54   "SessionType": 0,
55   "IsAnonymous": false,
56   "PublishedDesktopId": 20,
57   "CreatedDate": "2024-07-19T13:32:30.267Z",
58   "ModifiedDate": "2024-07-25T13:57:20.283Z"
59 }
60 }
61 ]
62 ]
63 }
```

Count by ConnectionFailureEnumValue

- With `groupby` and `aggregate` operations, you can get the `ConnectionFailureLogs` count grouped by `ConnectionFailureEnumValue`.

Request

```
1 GET /monitorodata/ConnectionFailureLogs?$apply=filter(CreatedDate gt
2   2024-07-01T00:00:00.00Z )/groupby((ConnectionFailureEnumValue),
3   aggregate(id with countdistinct as FailureCount)) HTTP/1.1
4 Host: api.cloud.com
5 Authorization: CWSAuth bearer=<Token>
6 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
      ConnectionFailureLogs",
11   "value": [
12     {
13
14       "@odata.id": null,
15       "ConnectionFailureEnumValue": 3,
16       "FailureCount": 63
17     },
18     {
19
20       "@odata.id": null,
21       "ConnectionFailureEnumValue": 4,
22       "FailureCount": 178
23     }
24   ]
25 }
26
27 }
```

Users with a high number of session failures

As part of this example, we query the ConnectionFailureLogs entity, where session failures are logged. The linking to the Users entity is done through the Sessions entity. We group by the user entity, get the distinct count of failures and select the top items.

Request

```
1 GET /monitorodata/ConnectionFailureLogs?$apply=filter(Session/StartDate
      ge 2025-05-01)/groupby((Session/User/FullName,
      ConnectionFailureEnumValue), aggregate(Id with countdistinct as
      FailureCount))&$orderby=FailureCount desc&$top=20 HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     ConnectionFailureLogs",
12     "value": [
13         {
14             "@odata.id": null,
15             "ConnectionFailureEnumValue": 100,
16             "FailureCount": 8,
17             "Session": {
18
19                 "@odata.id": null,
20                 "User": {
21
22                     "@odata.id": null,
23                     "FullName": "User Name"
24                 }
25             }
26         }
27     ]
28 }
29
30 ]
31 }
```

Application data request examples

June 2, 2025

Application represents a virtualized application running on a session.

Use Citrix Monitor Service API to get the details of applications.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{MonitorServiceHost}/Citrix/Monitor/OData/v4/Data/{ODataQuery}` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid bearer token.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/Applications` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

Active Applications with Desktop Group ID

- `LifecycleState=0` represents active applications.

Request

```
1 GET /monitorodata/Applications?$filter=LifecycleState eq 0&$expand=
  DesktopGroups($select=Id) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
    Applications",
11   "value": [
12     {
13
14       "Id": "40610fc4-cbc0-4458-baaa-07254ae54d41",
15       "Name": "Citrix Floor Plan Viewer",
16       "PublishedName": "Citrix Floor Plan Viewer",
17       "ApplicationType": 0,
18       "Enabled": false,
19       "AdminFolder": "",
20       "LifecycleState": 0,
21       "Path": "C:\\Program Files (x86)\\Norskale\\Norskale Agent
        Host\\VUEAppCmd.exe",
```

```
22     "BrowserName": "Citrix Floor Plan Viewer",
23     "CreatedDate": "2018-12-11T19:55:33.29Z",
24     "ModifiedDate": "2018-12-11T19:55:33.29Z",
25     "DesktopGroups": [
26         {
27             "Id": "0052e44f-f32d-476f-a307-cfd617baabda"
28         }
29     ]
30 }
31 ]
32 }
33 ]
34 }
35 }
```

Application count by ApplicationType

- Use `groupby` operation to get the count group by ApplicationType.

Request

```
1 GET /monitorodata/Applications?$apply=groupby((ApplicationType),
2   aggregate(Name with countdistinct as NumberofApplications)) HTTP/1.1
3 Host: api.cloud.com
4 Authorization: CWSAuth bearer=<Token>
5 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11       Applications",
12     "value": [
13         {
14             "@odata.id": null,
15             "ApplicationType": 0,
16             "NumberofApplications": 68
17         }
18     ]
19 }
```

```
18
19     ]
20 }
```

Active applications with count

- `LifecycleState eq 0` represents active applications.
- `$count=true` parameter appends the record count to the response.

Request

```
1 GET /monitorodata/Applications?$filter=LifecycleState eq 0&$count=true
   HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Applications",
11     "@odata.count": 68,
12     "value": [
13         {
14
15             "Id": "88040e8b-8b62-4cde-bc2c-014d87518191",
16             "Name": "MyCigna",
17             "PublishedName": "MyCigna",
18             "ApplicationType": 0,
19             "Enabled": false,
20             "AdminFolder": "",
21             "LifecycleState": 0,
22             "Path": "C:\\Program Files (x86)\\Norskale\\Norskale Agent
   Host\\VUEMAppCmd.exe",
23             "BrowserName": "MyCigna",
24             "CreatedDate": "2018-12-11T19:56:18.177Z",
25             "ModifiedDate": "2018-12-11T19:56:18.177Z"
26         }
   ]
}
```

```

27
28     ]
29 }

```

Application instance data request examples

June 2, 2025

Application instance is the record of an instance of a running application.

Use Citrix Monitor Service API to get the details of application instance.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{ MonitorServiceHost } /Citrix/Monitor/OData/v4/Data/{ ODataQuery }` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/ApplicationInstances` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

ApplicationInstances for non-terminated sessions

- `EndDate` is **null** and `ConnectionState` != 3 represent not terminated sessions.
- With the `expand` operator we enrich the output records with Application, Session and Machine details

Request

```

1 GET /monitorodata/ApplicationInstances?$filter=EndDate eq null and
    Session/ConnectionState ne 3&$expand=Application($select=Name,
    PublishedName,Path),Session($select=StartDate,ConnectionState,
    SessionType;$expand=Machine($select=Name))&$count=true HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     ApplicationInstances",
12     "value": [
13         {
14             "SessionKey": "559dfdb2-235f-4131-925c-21ca3983dcd5",
15             "StartDate": "2024-11-28T15:08:50.4433333Z",
16             "EndDate": null,
17             "Application": {
18
19                 "Name": "Paint",
20                 "PublishedName": "Paint",
21                 "Path": "C:\\Windows\\system32\\mspaint.exe"
22             }
23         },
24         "Session": {
25
26             "StartDate": "2024-11-28T15:08:21.0333Z",
27             "ConnectionState": 5,
28             "SessionType": 1,
29             "Machine": {
30
31                 "Name": "LAB\\TEST-MVDA-01"
32             }
33         },
34         "User": {
35
36             "Upn": "user.name@citrix.com",
37             "FullName": "User Name"
38         }
39     }
40 }
41
42     }
43
44 ]
45 }
```

ApplicationInstances for target application with Machine and User information

- Use `filter` operation to get the details for target application.
- Use `expand` operation to get the related Machine and User Information.

Request

```
1 GET /monitorodata/ApplicationInstances?$apply=filter(ApplicationId eq 6
   b07220e-b7d2-4f8d-9e43-49bd85222553 and EndDate eq null)&$expand=
   Session($expand=Machine,User) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   ApplicationInstances",
11   "value": [
12     {
13
14       "Id": 13361311,
15       "ApplicationId": "6b07220e-b7d2-4f8d-9e43-49bd85222553",
16       "SessionKey": "762e418d-270f-4db1-bc7a-2eb3747825fb",
17       "StartDate": "2024-07-28T10:06:19.363Z",
18       "EndDate": null,
19       "CreatedDate": "2024-07-28T10:06:19.363Z",
20       "ModifiedDate": "2024-07-28T10:06:19.363Z",
21       "Session": {
22
23         "SessionKey": "762e418d-270f-4db1-bc7a-2eb3747825fb",
24         "StartDate": "2024-07-28T10:06:05.397Z",
25         "LogOnDuration": 15408,
26         "EndDate": null,
27         "ExitCode": 1,
28         "FailureId": null,
29         "FailureDate": null,
30         "ConnectionState": 5,
31         "SessionIdleTime": null,
32         "ConnectionStateChangeDate": "2024-07-28T10:06:17.55Z",
```

```
33     "LifecycleState": 0,  
34     "CurrentConnectionId": 33952565,  
35     "UserId": 103038,  
36     "MachineId": "b5630f84-9234-485c-bb3b-68217feb451b",  
37     "SessionType": 1,  
38     "IsAnonymous": false,  
39     "PublishedDesktopId": null,  
40     "CreatedDate": "2024-07-28T10:06:17.55Z",  
41     "ModifiedDate": "2024-08-02T08:08:38Z",  
42     "Machine": {  
43  
44         "Id": "b5630f84-9234-485c-bb3b-68217feb451b",  
45         "Sid": "S-1-5-21-420551719-245851362-9522986-884824",  
46         "Name": "XXXX\\ADEAPPRB2D048",  
47         "DnsName": "test.wa.test.org",  
48         "LifecycleState": 0,  
49         "IPAddress": "10.10.201.250",  
50         "HostedMachineId": "423fd007-86ff-7729-96bf-f4cd05646e0a",  
51         "HostingServerName": "test.test.org",  
52         "HostedMachineName": "test",  
53         "IsAssigned": false,  
54         "IsInMaintenanceMode": true,  
55         "IsPendingUpdate": false,  
56         "AgentVersion": "1912.0.7000.7052",  
57         "AssociatedUserFullNames": "auto1005255",  
58         "AssociatedUserNames": "test\\auto1005255",  
59         "AssociatedUserUPNs": "auto1005255@test.org",  
60         "CurrentRegistrationState": 1,  
61         "RegistrationStateChangeDate": "2024-07-26T05:00:32.447Z",  
62         "LastDeregisteredCode": 300,  
63         "LastDeregisteredDate": "2024-07-19T07:38:43.81Z",  
64         "CurrentPowerState": 3,  
65         "CurrentSessionCount": 1,  
66         "ControllerDnsName": "test.local",  
67         "PoweredOnDate": null,  
68         "PowerStateChangeDate": "2024-07-24T10:09:09.293Z",  
69         "FunctionalLevel": 7,  
70         "FailureDate": null,  
71         "WindowsConnectionSetting": 0,  
72         "IsPreparing": false,  
73         "FaultState": 1,  
74         "OSType": "Windows 2019",  
75         "CurrentLoadIndexId": 258229225,  
76         "CatalogId": "693997ce-3024-4116-9b25-964a0b2f5f66",  
77         "DesktopGroupId": "1377ad1d-b245-4782-b7b4-45b093ae1fe5",  
78         "HypervisorId": "5ac693e1-2fb4-4293-97f2-3c4338e0aff0",  
79         "LastPowerActionType": 4,
```

```

80         "LastPowerActionReason": 5,
81         "LastPowerActionFailureReason": 400,
82         "LastPowerActionCompletedDate": "2024-07-26T04
           :59:41.257Z",
83         "Tags": [],
84         "Hash": "
           B136809A5709E6F020D3C5095A4FAC5C1F31AA4B5C78064C3F7102E56
           ",
85         "MachineRole": 0,
86         "CreatedDate": "2024-01-26T23:25:41.88Z",
87         "ModifiedDate": "2024-07-31T00:08:38.86Z"
88     }
89 ,
90     "User": {
91         "Id": 103038,
92         "Sid": "S
           -1-5-21-1518083187-624064714-1050887974-125084",
93         "Upn": "auto1005255@ak.test.org",
94         "UserName": "auto1005255",
95         "FullName": "auto1005255",
96         "Domain": "test",
97         "CreatedDate": "2023-07-12T04:43:39.47Z",
98         "ModifiedDate": "2023-07-12T04:43:39.47Z"
99     }
100 }
101 }
102 }
103 }
104 }
105 ]
106 ]
107 }

```

User data request examples

June 2, 2025

Users represent the users that have launched a session in the site.

Use Citrix Monitor Service API to get the details of users.

NOTE: The request examples of this section target **cloud** setups. However the OData queries also apply to **on-premises** setups. Simply adjust the request URL to be of the form `http[s]://{MonitorServiceHost}/Citrix/Monitor/OData/v4/Data/{ODataQuery}` and change the authentication to NTLM.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have the `bearer` token.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/Users`, for your region specific API gateway endpoint, see [Supported API Gateway endpoints](#).
- Invoke the API described in this document from a client host.

Users of target domain

- `Domain eq 'CITRITE'` represents the target domain.

Request

```
1 GET /monitorodata/Users?$filter=Domain eq 'CITRITE' HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11     Users",
12     "value": [
13         {
14             "Id": 2,
15             "Sid": "S-1-5-21-1076320343-3137401123-2908695819-83261",
16             "Upn": "matthew.bennett@citrix.com",
17             "UserName": "matthewben",
18             "FullName": "Matthew Bennett",
19             "Domain": "CITRITE",
20             "CreatedDate": "2018-12-12T11:30:57.15Z",
21             "ModifiedDate": "2021-05-10T12:27:18.223Z"
22         }
23     ],
24     {
25
```

```

26         "Id": 3,
27         "Sid": "S-1-5-21-1076320343-3137401123-2908695819-83263",
28         "Upn": "adm1_matthewben@citrix.com",
29         "UserName": "adm1_matthewben",
30         "FullName": "Matthew Bennett (ADM1)",
31         "Domain": "CITRITE",
32         "CreatedDate": "2018-12-12T11:34:37.637Z",
33         "ModifiedDate": "2021-07-26T15:36:01.04Z"
34     }
35 ]
36 }
37 }

```

Users with high logon duration

As part of this example, we query for the Sessions entity, which includes the logon duration values. The logon duration threshold can be adjusted. Session entity provides linking with the Users entity, as well as Machines and Connections, allowing us to enrich the result with such information (e.g. Logon Duration breakdown)

- `StartDate gt 2025-05-01` selects sessions started after the timestamp.
- `LogonDuration gt 120000` selects users with logon duration greater than 120 seconds.
- Using the `$expand` operator we enrich the output with Machine, DesktopGroup, User and Connection details.
- `orderby=LogOnDuration desc` sorts the results in descending order.

Request

```

1 GET /monitorodata/Sessions?$filter=StartDate ge 2025-05-01 and
  LogonDuration gt 120000 &$expand=Machine($select=Name;$expand=
  DesktopGroup($select=Name)),User($select=FullName,Upn),Connections(
  $select=LogOnStartDate,LogOnEndDate,BrokeringDuration,
  VMStartStartDate,VMStartEndDate,HdxStartDate,HdxEndDate,
  AuthenticationDuration,GpoStartDate,GpoEndDate,LogOnScriptsStartDate
  ,LogOnScriptsEndDate,ProfileLoadStartDate,ProfileLoadEndDate,
  InteractiveStartDate,InteractiveEndDate,ClientName,ClientPlatform,
  Protocol,IsReconnect;$filter=IsReconnect eq false),&$select=
  StartDate,SessionKey,SessionType,LogonDuration&$orderby=
  LogOnDuration desc HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
11         Sessions",
12     "value": [
13         {
14             "SessionKey": "f533b450-57eb-4ae1-89d6-393a02872e6c",
15             "StartDate": "2024-11-26T15:06:55.157Z",
16             "LogOnDuration": 146957,
17             "SessionType": 0,
18             "Machine": {
19
20                 "Name": "TEST\\LAB3-WPD216",
21                 "DesktopGroup": {
22
23                     "Name": "LAB3-W10-P"
24                 }
25             }
26         },
27     ],
28     "User": {
29
30         "Upn": "user.name@citrix.com",
31         "FullName": "User Name"
32     },
33     "Connections": [
34         {
35
36             "ClientName": "CSGPROD152994",
37             "ClientPlatform": "Windows",
38             "IsReconnect": false,
39             "Protocol": "HDX",
40             "LogOnStartDate": "2024-11-28T07:34:58.353Z",
41             "LogOnEndDate": "2024-11-28T07:39:32.057Z",
42             "BrokeringDuration": 24,
43             "VMStartStartDate": "2024-11-28T07:35:29.727Z",
44             "VMStartEndDate": "2024-11-28T07:36:56.51Z",
45             "HdxStartDate": "2024-11-28T07:37:15.893Z",
46             "HdxEndDate": "2024-11-28T07:37:17.6Z",
47             "AuthenticationDuration": 1484,
48             "GpoStartDate": "2024-11-28T07:39:27.71Z",
49             "GpoEndDate": "2024-11-28T07:39:33.353Z",
50             "LogOnScriptsStartDate": null,
51             "LogOnScriptsEndDate": null,
```

```

53         "ProfileLoadStartDate":"2024-11-28T07:37:24.907Z",
54         "ProfileLoadEndDate":"2024-11-28T07:39:27.703Z",
55         "InteractiveStartDate":"2024-11-28T07:39:27.703Z",
56         "InteractiveEndDate":"2024-11-28T07:39:32.057Z"
57     }
58
59 ]
60 }
61
62 ]
63 }

```

Users with high ICARTT (low responsiveness)

As part of this example, we query for the SessionMetrics entity, which includes the ICARTT values. The linking to the Users entity is done through the Sessions entity.

Request

```

1 GET /monitorodata/SessionMetrics?$filter=CollectedDate ge 2025-05-01
   and IcaRttMS gt 300&$select=IcaRttMS&$expand=Session($select=
   StartDate,SessionKey,SessionType;$expand=User($select=FullName,Upn),
   Machine($select=Name),CurrentConnection($select=ClientName,
   ClientPlatform,Protocol))&$orderby=IcaRttMS desc HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10     "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   Sessions",
11     "value": [
12         {
13
14             "IcaRttMS":40203,
15             "Session":{
16

```

```
17     "SessionKey":"b28f7628-852a-4865-a1d2-2f7ab0a08310",
18     "StartDate":"2024-11-21T16:18:20.123Z",
19     "SessionType":0,
20     "User":{
21         "Upn":"user.name@citrix.com",
22         "FullName":"User Name"
23     },
24     "Machine":{
25         "Name":"TEST\\LAB3-WPD216"
26     },
27     "CurrentConnection":{
28         "ClientName":"HTML-3420-2374",
29         "ClientPlatform":"HTML5",
30         "Protocol":"HDX"
31     }
32 }
33 ]
34 }
```

Cost optimization data request examples

June 2, 2025

The MachineCostSavingsSummaries entity provides aggregated metrics related to the cost savings applied on the VDAs.

NOTE: The request examples of this section target **cloud** setups.

Prerequisites

- Read the [Get started with Citrix Cloud APIs](#) section to ensure that you have a valid **bearer token**.
- Use the endpoint `https://{ ApiGatewayEndpoint } /monitorodata/MachineCostSaving` for your requests. See [Supported API Gateway endpoints](#) for a list of region specific API gateway endpoints.
- Invoke the API described in this document from a client host.

Find out total savings

For more information on cost savings, see the [Cost savings](#) page.

- Apply `aggregate` operation to calculate the sum of the saved amount.

Request

```
1 GET /monitorodata/MachineCostSavingsSummaries?$apply=aggregate(
   TotalAmountSaved with sum as TotalAmountSavedSum) HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 556
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10    "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
   MachineCostSavingsSummaries(TotalAmountSavedSum)",
11    "value": [
12      {
13
14          "@odata.id": null,
15          "TotalAmountSavedSum": 2035677.9166666665
16      }
17    ]
18  }
```

Find out the machine state

- Filter based on the `SummaryDate` and select the fields of interest.

Request

```
1 GET /monitorodata/MachineCostSavingsSummaries?$filter=SummaryDate eq
  2025-05-09T06:00:00Z&$select=TotalMachinesCount,
  PowerManagedMachinesOff,PowerManagedMachinesInHibernate,
  PowerManagedMachinesOnWithSessions,
  PowerManagedMachinesOnWithoutSessions HTTP/1.1
2 Host: api.cloud.com
3 Authorization: CWSAuth bearer=<Token>
4 Citrix-CustomerId: <CustomerId>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Thu, 01 Aug 2024 06:23:00 GMT
3 Content-Type: application/json; odata.metadata=minimal
4 Content-Length: 5506
5 Connection: keep-alive
6 odata-version: 4.0
7
8 {
9
10   "@odata.context": "https://api.cloud.com/monitorodata/$metadata#
  MachineCostSavingsSummaries(TotalMachinesCount,
  PowerManagedMachinesOff,PowerManagedMachinesInHibernate,
  PowerManagedMachinesOnWithSessions,
  PowerManagedMachinesOnWithoutSessions)",
11   "value": [
12     {
13
14       "TotalMachinesCount": 30,
15       "PowerManagedMachinesOff": 6,
16       "PowerManagedMachinesInHibernate": 0,
17       "PowerManagedMachinesOnWithSessions": 0,
18       "PowerManagedMachinesOnWithoutSessions": 24
19     },
20     {
21
22       "TotalMachinesCount": 14,
23       "PowerManagedMachinesOff": 4,
24       "PowerManagedMachinesInHibernate": 0,
25       "PowerManagedMachinesOnWithSessions": 9,
26       "PowerManagedMachinesOnWithoutSessions": 1
27     }
28   ]
29 }
30 ]
31 }
```

Other examples relevant to Cost Optimization

- [Find unused VDAs](#)
- [Find the daily average uptime in minutes](#)
- [Find failed machine power action](#)

Resources

June 2, 2025

Getting Started Guides

- [Citrix Monitor service API documentation](#)
- [Get Citrix Cloud authentication credentials for OData](#)
- [Learn more about accessing Citrix Monitor OData v4](#)

Citrix Monitor Postman Collection

- [Public Citrix Postman workspace](#)
- [Monitor OData API examples collection](#)
- [Monitor OData environment](#)

Do More With Data Sets

- [Data models-monitor service schema](#)
- [OData API and PowerShell SDK tutorial \(video\)](#)
- [Monitoring Citrix Cloud with OData and PowerShell \(video\)](#)
- [Monitoring CVAD with OData \(video\)](#)
- [Learn more about the OData syntax](#)
- [Descriptions for enums and error codes](#)

Developer Community

- [PowerBI template](#)
- [PowerBI step-by-step integration](#)
- [Powershell data collection scripts](#)
- [API usage samples and use cases \(GitHub\)](#)

Blogs

- [Deep dive into Monitor API](#)

Release notes

December 10, 2024

January 2025

- Add the default filter in [API Exploration](#).
- Add [How to access On-Premises Monitor OData API](#).
- Update the PowerQuery template.
- Add [How to invoke API](#).

October 2024

- Add SpaApplications, SpaEnumerations and SpaSessions for SPA.

September 2024

- Re-organize the document.
- Add the [How to](#) section for use cases.
- Add the [Resources](#) section and included all the required information for Monitor OData.
- Add the [API Exploration](#) section for API usage.

May 2024

- Update the API doc for current release.

May 2023

- Update the API doc for 7-15-ltsr.



copyright-text-footer